SocialDrone: An Integrated Social Media and Drone Sensing System for Reliable Disaster Response

***Abstract*—Social media sensing has emerged as a new disaster response application paradigm to collect real-time observations from online social media users about the disaster status. Due to the noisy nature of social media data, the task of identifying trustworthy information (referred to as "truth discovery") has been a crucial task in social media sensing. However, existing truth discovery solutions often fall short of providing accurate results in disaster response applications due to the spread of misinformation and difficulty of an efficient verification in such scenarios. In this paper, we present SocialDrone, a novel closed-loop social-physical active sensing framework that integrates social media and unmanned aerial vehicles (UAVs) for reliable disaster response applications. In SocialDrone, signals emitted from the social media are distilled to drive the drones to target areas to verify the emergency events. The verification results are then taken back to improve the sensing and distillation process on social media. The SocialDrone framework introduces several unique challenges: i) how to drive the drones using the unreliable social media signals? ii) How to ensure the system is adaptive to the high dynamics from both the physical world and social media? iii) How to incorporate real-world constraints (e.g., the deadlines of events, limited number of drones) into the framework? The SocialDrone addresses these challenges by building a novel integrated social-physical sensing system that leverages techniques from game theory, constrained optimization, and reinforcement learning. The evaluation results on a real-world disaster response application show that SocialDrone significantly outperforms state-of-the-art truth discovery schemes and drone-only solutions by providing more rapid and accurate disaster response. To the best of our knowledge, SocialDrone is the first solution that integrates social media sensing with drone-based physical sensing systems for disaster response applications.**

I. Introduction

Social media sensing has recently emerged as a new application paradigm where social media posts from online users are parsed and analyzed to report the status of the physical world [1]. A critical application enabled by social media sensing is disaster response. During a disaster event, social media platforms often provide real-time and valuable information of emergency events. Figure 1 illustrates an example of two geo-tagged tweets posted during the 2018 California Camp Fire, one of which talks about a road closure and the other about a missing person. An important challenge of using social media for disaster response is *truth discovery* where the goal is to identify trustworthy information contributed by massive unvetted sources from online social media [2]. While great efforts have been made on developing reliable truth discovery solutions [3]–[6], several limitations still prevent these solutions from being applied in disaster response applications. In particular, the truth discovery algorithms are known to underperform in the presence of widespread misinformation, which is common during disaster scenarios [7]. The key to address this issue is to acquire ground truth to validate the source reliability and event correctness. However, such ground truth requires a significant amount of manual effort and is delay prone during the real-world disaster events [8].

Figure 1. Tweets posted during the 2018 California Camp Fire

This paper focuses on a new disaster response application where the social media (e.g., Twitter, Instagram) and unmanned aerial vehicles (UAVs) are used together to obtain real-time situation awareness in the aftermath of a disaster. The mobility and agility of UAVs allow them to be quickly sent to the disaster events to collect real-time evidence and verify whether the events are actually happening before sending out rescue teams [9]. The reliable and high quality measurements provided by the drones naturally complement the uncertain estimation of the social media sensing. In fact, the drones can verify various disaster events (e.g., fires, floods, leakage of toxic chemicals) with high confidence by leveraging dedicated sensors and advanced machine learning techniques [10]–[12]. However, one factor that limits the feasibility of UAVs is that they require a great degree of manual inputs from human operators to be narrowed-down to the location of disaster events. This may give rise to delayed responses that are unacceptable in disaster response applications [13]. Furthermore, a drone is an expensive resource that has to be utilized sparingly in the physical world. It is not practical to launch an arbitrary number of drones for "scouting" a large area. In contrast to UAVs, the outreach of social media sensing is far greater (e.g., anyone possessing a portable device with Internet connectivity can report an incident during a disaster).

We pose a challenging research question in this paper: *could the information from social media be used to drive drones to provide faster and more effective responses during the course of a disaster?* For example, if the tweets shown in Figure 1 could be used to drive a set of drones, it could potentially expedite the search and rescue operations of missing people and identify the condition of road closures during the Camp Fire event. However, a few important technical challenges need to be carefully addressed before the social media sensing and drone system can act in tandem.

The first challenge lies in leveraging the noisy and uncertain "signals" emitted from the social media to drive the drones to the desirable locations. While state-of-the-art truth discovery solutions have decent performance in identifying credible disaster events, they are only as good as the information presented on social media. None of the current solutions integrate with any physical component like drones to actively verify the information they estimate and improve the estimation accuracy [5], [14]. Therefore, it remains to be an open question on how to leverage the unreliable social media signals to reliably control the drones for effective disaster response.

The second challenge lies in developing a closed-loop system that seamlessly integrates the social media sensing and the drone sensing paradigms. The design of such a closed-loop system requires a careful control of the interactions between the social and physical worlds, both of which are highly dynamic. For example, the number of users on social media and their reliability may change drastically over time [3]. Similarly, the events in the physical world occur at different locations and the truthfulness of the events may also evolve [6]. It entails a tight integration between the social media sensing and drone system by using the social media signals to drive the drones and afterwards utilizing the obtained information from drones to improve the performance of social media sensing. This closed-loop challenge at the intersection of cyber, physical and social spaces has not been addressed in the current literature from both social media and drone systems.

The last challenge is the difficulty imposed by several constraints from the physical world (e.g., the resource constraints of the drones and the deadline constraints imposed by the events). In particular, we assume only a finite number of drones can be utilized for a particular disaster response application at any given time. Furthermore, we also assume the identified events in a disaster have certain *deadlines* that reflect their urgency, after which it may not be viable to verify them. For instance, if there is a report of a person being injured during a disaster situation, that event often has a tight deadline. We note that it is a known NP-hard problem of finding an optimal allocation strategy to assign a limited number of drones to events with different deadlines [15], [16]. However, our problem is even more challenging because we also need to consider the dynamics of the social and the physical worlds as well as the interactions between them.

In this paper, we develop a closed-loop active sensing framework, SocialDrone (Figure 2), that integrates the social media sensing with the drones for reliable disaster response. Our system consists of a new game-theoretic drone task allocation module to selectively choose the desired locations to send to drones and verify the event information extracted from unreliable social media data. A path planning module is developed to ensure the flight trajectory of the drones meet deadline constraints of the target events. Finally, we design a closed-loop source selection module that leverages the data collected from drones to dynamically filter out unreliable users on social media to calibrate the accuracy of the social media sensing and eventually improve the effectiveness of the disaster response application.

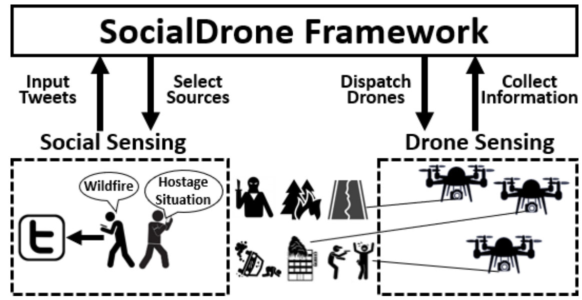

Figure 2. An overview of the SocialDrone framework

To the best of our knowledge, the SocialDrone framework is the first solution that integrates social media sensing with drone-based sensing for reliable disaster response applications. We evaluated our framework in comparison to both social media based and drone based disaster response systems on a real world dataset collected from Twitter during a natural disaster: California Camp Fire in November 2018. The results show that SocialDrone significantly outperforms the compared baselines in providing more accurate (15% increase in F1-score) and timely information (10% increase in deadline hit rate) during a disaster event.

II. Related Work

A. Truth Discovery in Social Media Sensing

A key challenge in social media sensing applications is the data reliability issue [1]. Various truth discovery models have been developed to address this challenge. For example, Yin *et al.* developed *Truth Finder*, a probabilistic algorithm using iterative weight updates [4]. Wang *et al.* proposed a scheme that jointly estimates source reliability and claim correctness using the maximum-likelihood estimation approach [5]. Li *et al.* proposed a confidence-aware truth discovery method to automatically detect truths from data with long-tail phenomenon [17]. A comprehensive survey of truth discovery schemes is presented in [18]. One major limitation of these truth discovery schemes is that they solely rely on the social media data that tend to be noisy, sparse and unreliable. The uncertain outputs of these solutions can lead to significant false alarms in disaster response and mislead the rescue teams to areas where no emergency exists [7]. In contrast, our SocialDrone framework integrates the social media with drones to address the data reliability challenge of social media sensing.

B. Disaster Response with Physical Sensing

Disaster response is a crucial application to ensure immediate resolution to emergent and hazardous events [19]. A critical step in disaster response is to identify the area where a disaster has taken place and determine the severity of the damage caused by a disaster. Several disaster assessment systems have been built that make use of drones flying to the location of disaster events. For example, Zander *et al*. built a smart emergency response system in which survivors tag themselves using their handheld devices and rescuers dispatch drones to find them [20]. Alazawi *et al.* proposed a disaster management system that utilizes vehicular networks in urban environments to gather and distribute the information in the aftermath of a disaster [21]. However, the above approaches primarily rely on the signals from human operators, which could be both slow and limited in scope. Moreover, the chance of discovering an event is subject to the drones patrolling in close vicinity of the events, which may be unlikely when the number of drones is limited in an application. In contrast, SocialDrone framework automatically drives the drones to the desirable locations and recovers the truthful state of the events based on the signals emitted from social media.

C. Active Learning

In an active learning framework, a learning algorithm actively obtains labels of the selected instances from domain experts [22]. For example, Wang *et al.* developed an active learning framework for crowdsensing-based air quality monitoring where the application leverages a small subset of available workers to selectively collect air quality measurements of assigned areas [23]. Zhang *et al.* developed an adaptive sampling based active learning framework to selectively choose most meaningful social media feeds to perform online topic detection [24]. Ambati *et al.* applied active learning techniques to dynamically query the crowd for annotations of texts and use the annotations to train an AI model to translate low resource languages [25]. Our work is clearly different from existing work in two aspects: 1) we are the first that use drones to do the active sensing and consider the physical constraints and costs of using drones; 2) we designed a novel source selection mechanism dedicated to improving the performance of truth discovery algorithms in social media sensing.

III. Problem Formulation

In this section, we first present the basic terms and assumptions of our model and then define the objective of our problem. In a disaster response application, we consider a physical region of interest with a specific duration of sensing. To access the scope of the sensing process at a reasonable granularity, we divide the sensing region into distinct *sensing cells* that are clustered into a sensing grid and distribute the duration of sensing across *sensing cycles*.

DEFINITION 1. Sensing Cycles: We divide the duration of an application into L fixed intervals called *sensing cycles*, where $l \in [1, L]$ denotes the l^{th} sensing cycle.

In our problem, we consider a scenario where a group of M_l social media users report a collection of N_l distinct events, $\mathcal{E} = (E_{l,1}, E_{l,2}, ..., E_{l,N_l})$, at sensing cycle l. The events can be any significant occurrence taking place in the course of a natural or man-made disaster. For example, an event can be a person trapped under a tree log, a road blocked due to an accident, a suspect evading from a crime scene, or a possible bomb. We define an event $E_{l,p}$ to be a binary variable:

DEFINITION 2. Event $E_{l,p}$:

$$E_{l,p} = \begin{cases} 1, & \textit{if reported event } E_{l,p} \textit{ exists} \\ 0, & \textit{if reported event } E_{l,p} \textit{ does not exist} \end{cases} \tag{1}$$

We also denote $\widehat{E_{l,p}}$ to indicate the estimated truth of event $E_{l,p}$ by our SocialDrone system.

We formally define a *task* as follows:

DEFINITION 3. The Task for Drone: A task for drones at a sensing cycle refers to the location of an event where the drone should be sent to.

We define the data from the social media (e.g., tweets from Twitter) as follows:

DEFINITION 4. Social Media Data $\mathcal{S}$: the social media posts that report events in the physical world during a disaster. An example of a social media report is shown in Figure 1.

We also define a set of G drones, $\mathcal{D} = (D_1, D_2, ..., D_G)$ that will be triggered and dispatched to the cells of interest after analyzing the events obtained from the social media. Figure 3 presents an illustrative example of the concepts defined above.

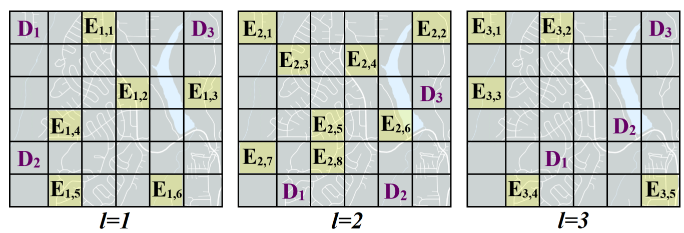

Figure 3. Snapshot of the sensing grid across subsequent sensing cycles. When $l = 1$, $E_{1,1}, E_{1,2}, ...$ indicate the events occurring in sensing cycle 1. $D_1, D_2, ...$ denote the present locations of the drones.

We further define a few key attributes of the event $E_{l,p}$.

DEFINITION 5. Event Deadline $\delta_{l,p}$ for event $E_{l,p}$: an event is assumed to have a deadline based on its urgency. For example, the event of an injured person will typically have a tighter deadline than the event of a road closure.

The event deadline is a critical constraint in our problem, i.e., the drones must reach the event location before the deadline expires in order to verify whether the event actually happens. We assume that the deadlines of events are shorter than the duration of the sensing cycle. For the event with a longer deadline than the sensing cycle, we split the event into multiple events. Each of the resulting events will have a deadline shorter than the sensing cycle. The split events also inherit the *priority* from the original event to avoid the potential problem of "priority inversion" [26].

Using the above definitions, we therefore define the goal of the SocialDrone framework. Given the social media data inputs $\mathcal{S}$, a set of drones $\mathcal{D}$, and corresponding deadlines for events $\delta_{l,p}$, the objective of the SocialDrone framework is to minimize the discrepancy between the estimated validity of the events and their ground truth by solving a constrained optimization problem as follows:

$$\underset{\widehat{E_{l,p}}}{\arg\min} \sum_{l=1}^{L} \sum_{P=1}^{N_l} (abs(\widehat{E_{l,p}} - E_{l,p}) | \delta_{l,p}, \mathcal{D}, \mathcal{S}) \qquad (2)$$

where abs is a function to generate the absolute value of a given number.

IV. Solution

In this section, we present the SocialDrone framework that integrates the social media sensing and the drone sensing systems to address the problem formulated in the previous section. The SocialDrone is essentially an active sensing scheme that selectively queries a subset of sensing cells by dispatching drones driven by social media signals, and leverages the acquired measurements to improve the overall accuracy of the sensing system. SocialDrone incorporates three major components: i) a reliable social signal distillation (RSSD) module that distills reliable signals from noisy social media data to dispatch drones; ii) a drone task allocation (DTA) module that allocates a subset of events to the drones based on the distilled social signals and guides the drones to the location of the events before their deadlines expire; iii) a trustworthy source selection (TSS) module that leverages measurements obtained by the drones to discard unreliable social sources and improve the quality of signals from social media. An overview of SocialDrone is shown in Figure 4.

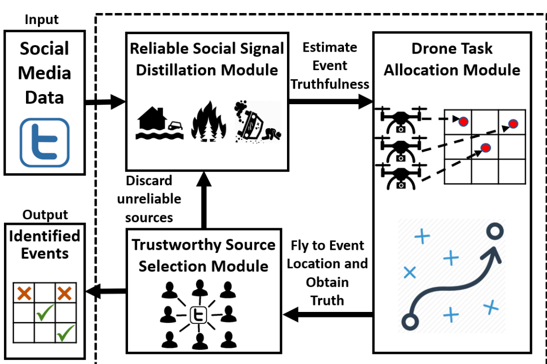

Figure 4. Architecture of the SocialDrone Framework

A. Reliable Social Signal Distillation (RSSD) Module

The social media posts are often generated by unvetted users with unknown reliability. Several *truth discovery* algorithms have been developed to assess the truthfulness of the event as well as the reliability of users [5], [14]. The RSSD module incorporates a set of state-of-the-art truth discovery solutions into our framework to distill useful signals from social media.

In particular, we leverage an ensemble of a diverse set of representative truth discovery schemes to jointly identify: 1) the estimated label of the event; and 2) the uncertainty of the estimated label. The rational of the ensemble technique is to eliminate the bias of each individual truth discovery algorithm by combining them into a more robust and accuracy scheme [27].

Formally, we choose a diverse set of M representative truth discovery algorithms $TD_1, TD_2, ..., TD_M$. We refer to these algorithms as an "ensemble" and each of the algorithm as a "ensemble member". The inputs to each truth discovery scheme are the social media signals (e.g., the tweets) and the algorithm outputs two sets of scores defined below. The outputs of all the members jointly decide the final label of the events.

DEFINITION 6. Event Correctness $C_{l,p}^{(m)}$ for event $E_{l,p}$ from member TD_m: A score in the range (0,1] that signifies the possibility that an event is true. The higher the value of $C_{l,p}^{(m)}$ is, the more likely the event $E_{l,p}$ is true (i.e., $E_{l,p}$ exists).

DEFINITION 7. User Reliability $\mu_u^{(m)}$ for user $U_u^{(m)}$: A score in the range $(0,1)$ that represents trustworthiness of a social media user. Intuitively, the higher value of $\mu_u^{(m)}$ is, the greater likelihood user U_u contributes a valid report.

For each member TD_m, we define a weight w_m^l, that represents the authority level of the algorithm in determining the final label of the event. The higher the weight, the more trustworthy the algorithm's output is.. The expert weight is dynamically adjusted based on the performance of each expert. We use a feedback control mechanism to update the weights by comparing the estimated labels of each expert with the actual ground truth collected by the drones [28].

Given the expert weights and outputs of the members, the ensemble decides the final label of the event, which is the weighted sum of the label distributions of all ensemble members. Formally, we calculate the event $E_{l,p}$'s truthfulness (denoted as $C_{l,p}$), and source reliability for user U_u (denoted as μ_u)as:

$$C_{l,p} = \sum_{m=1}^{M} w_m^l \times C_{l,p}^{(m)}, \quad \mu_u = \sum_{m=1}^{M} w_m^l \times \mu_u^{(m)} \qquad (3)$$

The event truthfulness score $C_{l,p}$ is used to infer the label of events where we do not dispatch drones. The user reliability score μ_u is used later for closed-loop control to the RSSD module via a source selection mechanism. The details are presented in Section IV-C.

We then quantify the uncertainty of the estimated event truthfulness. We define a new metric called *event ambiguity* that measures the members' overall uncertainty of labeling a event's truthfulness. The intuition is that, if the members have high disagreement, then the system cannot confidently identify the truthful label of the event. In that case, we assign it higher priority for drones to verify. To this end, we derive the event ambiguity score $\mathcal{H}_{l,p}$ as:

$$\mathcal{H}_{l,p} = -\sum_{m=1}^{M} \Big(Pr(C_{l,p}^{(m)}) \times \log Pr(C_{l,p}^{(m)}) + Pr(1 - C_{l,p}^{(m)}) \times \log(1 - Pr(C_{l,p}^{(m)}) \Big) \times \mathrm{Var}(C_{l,p}^{(1)}, .., C_{l,p}^{(m)}, ..., C_{l,p}^{(M)}) \qquad (4)$$

where Var$(\cdot)$ calculates the variance of a sequence. In the above formulation, the first term $-\sum_{m=1}^{M} \Big(Pr(C_{l,p}^{(m)}) \times$

$\log Pr(C_{l,p}^{(m)}) + Pr(1 - C_{l,p}^{(m)}) \times \log(1 - Pr(C_{l,p}^{(m)}))\Big)$ captures the aggregation of internal ambiguity of each member, characterized by entropy. The second term $\text{Var}(C_{l,p}^{(1)}, .., C_{l,p}^{(m)}, ..., C_{l,p}^{(M)})$ defines the global ambiguity of all members in terms of the variance of their output truthfulness scores.

B. Drone Task Allocation (DTA) Module

The DTA module is designed to leverage the outputs of the RSSD module to drive the drones to the desirable locations and actively collect ground truth information of disaster events. The DTA module is a two-step process: 1) a game theoretic task assignment scheme is developed to identify the optimal event locations as well as the sequences of event locations to visit for each drone; 2) a path planning scheme is developed to identify the actual trajectory of visiting the assigned event locations for each drone. We discuss the details of the two schemes below.

1) Game Theoretic Task Assignment: The goal of the game theoretic task assignment is multi-fold. First, due to the limited number of drones, the scheme must carefully choose the subset of events to verify by the drones which helps to augment the performance of RSSD module. Second, the physical constraints (e.g., the velocity of the drones and the deadlines of the events) must be considered. However, the above two aspects of the goal can sometimes be at odds with each other. For example, an event whose ground truth label could help significantly improve the accuracy of RSSD may be too far away from the drones to reach before its deadline expires. Moreover, it may not be optimal to send multiple drones to the same or nearby locations and cause contention between drones. To address the above challenges, we develop a bottom-up game theoretic (BGT) drone task allocation scheme to assign tasks to the event location.

In game theory, congestion games are often used to mitigate resource contention (e.g., event locations) among a set of players (e,g., drones). We adopt a particular subclass of congestion games called singleton weighted congestion games [29] in our task allocation module. In a singleton weighted congestion game, the expected payoff of each task monotonically decreases as the number of players (drones) that picked the task increases. We assume that each drone defines its own utility function (weighted congestion property) and only picks one task at a time (singleton property). We note that the Pure Strategy Nash Equilibrium is guaranteed to exist under the above singleton weighted congestion game protocol [30]. This property is crucial for the drones to make mutually satisfactory task allocation decisions.

There are two critical components in our singleton weighted congestion game protocol: *utility function* and *congestion rate*. We formally define them below.

DEFINITION 8. Utility Function for event $E_{l,p}$: it represents the benefit for picking a task (i.e., event location).

DEFINITION 9. Congestion Rate $\gamma_{l,p}$ for event $E_{l,p}$: A score in the range of $[1, G+1]$ that indicates the level of congestion. Specifically, $\gamma_{l,p}$ is the number of drones that pick event $E_{l,p}$ plus one. When a drone selects an event, the congestion rate for that event is incremented by one.

In our model, we devised a customized utility function for drone D_g, referred to as *event priority score* as follows:

$$u_{l,p}^g = \frac{\lambda_1 \mathcal{H}_{l,p} + \lambda_2 \omega_{l,p}^g + \lambda_3 \delta_{l,p}}{\gamma_{l,p}} \tag{5}$$

The above utility function prioritizes the events for drone task allocation based on three factors: i) the uncertainty of an event, as captured by the ambiguity score (i.e., $\mathcal{H}_{l,p}$)); ii) the Euclidean distance to the event from the drone, denoted as $\omega_{l,p}^g$; and iii) the deadline of the event, as captured by $\delta_{l,p}$. In particular, the uncertainty factor prioritizes the events reported by social media that the truth discovery algorithms are not confident about. The deadline factor is designed to prioritize the events with tighter deadlines. The distance factor prioritizes events with shorter distance from the drones for the sake of energy savings. λ_1, λ_2 and λ_3 represent the weights of each factor. Their values are computed using *proportional control*, a widely adopted feedback loop technique in control [31]. Finally, the congestion rate (i.e., $\gamma_{l,p}$) on the denominator of the utility function is designed to avoid contention of drones. In particular, if two drones pick the same event, the utility will decrease for both drones. Each drone then makes its best decision towards maximizing its utility, until a Nash Equilibrium is found. We note that each drone in the system can pick multiple events to explore based on the utility. The Nash Equilibrium (NE) exists in the proposed game where every drone is assumed to know the best strategies of all other drones (i.e., picking the task has the highest utility) and no drone has anything to gain by only changing its own selected tasks. For each sensing cycle, the congestion game is played multiple rounds until a NE is achieved. We leverage the *best-response dynamics* algorithm to find the NE [32]. The process for task allocation is summarized in Algorithm 1.

2) Path Planning: Once all the tasks are assigned to the drones by the BGT scheme, we develop a path planning scheme to carefully plan the routes for the drones to reach the assigned locations of the events. The objective of the path planning is twofold: 1) meet the deadline requirements of the events, and 2) minimize the energy consumption of drones. We particularly emphasize on reducing the energy usage because a drone can be assigned multiple tasks and it has a finite flight time constrained by the battery capacity. Our goal is to make the best effort to ensure the drones can reach all the assigned destinations before they run out of batteries. We elaborate our path planning algorithm below.

For each drone D_g, the physical dynamics is described by a discrete form:

$$w_g(t_{\iota+1}) = A_d w_g(t_\iota) + B_d f_g(t_\iota), \tag{6}$$

where $\iota \in \mathbb{N}$ is the sampling index. $w_g \in \mathbb{R}^4$ is the state of drone g with $w_g = [p_g \quad v_g]$, where $p_g, v_g \in \mathbb{R}^2$ are the position and velocity of the drone, respectively.

Algorithm 1 Singleton Congestion Game for Selecting Events

Input: $D, E, \mathcal{H}_{l,p}, \delta_l, p, \omega_l^g, p, \gamma_l, p, \lambda_1, \lambda_2, \lambda_3$
Output: Sets of tasks for drones, $\mathcal{F}_l = \{F_1^l, F_2^l, ..., F_G^l\}$ at sensing cycle l

Initialize $\mathcal{F}_l, \gamma$, $u_{l,p}^g$
for $l = 1 : L$ **do**
  while all *doubtful* events are not selected **do**
    for $g = 1 : G$ **do**
      for $p = 1 : N_l$ **do**
        if $\gamma_{l,p}$ does not exist **then**
          $\gamma_{l,p} = 1$
        end if
        compute $u_{l,p}^g$ based on Equation (5)
      end for
      $taskScores = u_{l,p}^g$
      for $p = 1 : N_l$ **do**
        $selectTask = maximum(taskScores)$
        p'= value of p for $maximum(taskScores)$
        if $selectTask$ not in F **then**
          $F_l^g = append(E_{l,p'})$
          $\gamma_{p'} = \gamma_{p'} + 1$
        else
          remove $selectTask$ from $tasks$
        end if
      end for
    end for
  end while
end for

$f_g = [f_{g,1} \quad f_{g,2}]^T \in \mathbb{R}^2$ is the local admissible control force. (A_d, B_d) are discretized system matrices with proper dimensions.

To generate feasible path leading to the selected location of events, we model the goals as small convex polygonal boxes characterized by $\{p \in \mathbb{R}^2 | a_{g,j}^T [I_2 \quad O_2] p + b_{g,j} \leq 0, j = 1, 2, \ldots, M_g\}$, where $a_{g,j} \in \mathbb{R}^2$, $b_{g,j} \in \mathbb{R}$, and $I_2, O_2 \in \mathbb{R}^{2\times 2}$ denote the 2-dimensional identity and zero matrices, respectively [33]. We assume that all drones share a synchronized clock [34]. The flight time $Y_{l,p}$ to the location of a selected event $E_{l,p}$ is upper-bounded by the event deadline $\delta_{l,p}$. The motion planning constraints can be summarized as:

$$a_{g,j}^T [I_2 \quad O_2] w_g(Y_{l,p}) + b_{g,j} \leq 0 \tag{7}$$

The motion planning constraint requires drone D_g to approach the selected location of an event within $Y_{l,p}$.

To minimize the energy consumption of drones, we define the following cost function $\Psi_{g,l,p}$ to represent the energy consumption with a goal penalty for drone g [35]. The goal penalty is used to push drones approaching the selected events.

$$\Psi_{g,l,p} = \sum_{\iota=1}^{Y_{l,p}} (q^T \times |w_{g,t_\iota}| + r^T \times |f_{g,t_\iota}|) + \sum_{\iota=1}^{Y_{l,p}} d_{g,\iota} \tag{8}$$

where q and r are non-negative weighting vectors and $|.|$ denotes the element-wise absolute value. The second term, the goal penalty, is defined as the Manhattan distance between the drone and the selected location of an event.

The path planning module aims to find the optimal path that minimizes energy cost given in Equation (8) while satisfying drone dynamics in Equation (6) as well as the task assignment and event deadline constraints in Equation (7). We solve the constrained optimization problem by encoding it as a mixed integer linear programming (MILP) problem. In particular, we adopt the approach used in [35] by introducing slack vectors with additional constraints such that $\Psi_{g,l,p}$ can be transformed into a linear cost function. The path planning problem can then be formulated as a MILP problem since both drone dynamics in Equation (6) and path planning constraints specified by Equation (7) are also linear. The MILP problem is solved using a commercial solver Gurobi [36].

C. Trustworthy Source Selection (TSS) Module

The TSS module leverages the sensing measurements collected by drones in DTA module to calibrate the truth discovery algorithms in the RSSD module and improve the overall system performance. In particular, the TSS module dynamically decides the threshold to filter out unreliable sources from social media. This process is particularly challenging due to several reasons. First, there exists a unique trade-off between the source selection and the performance of the truth discovery schemes. Specifically, if the system discards too few unreliable sources or does not filter sources at all, the false claims from unreliable sources will significantly degrade the system performance. However, if the system discards too many sources, the performance will then suffer from the data sparsity problem [17]. Second, the sources and the claims they contribute may both change significantly over time. It is not a trivial task to decide on the optimal threshold for source selection in such dynamic settings [7].

To address the source selection problem, we develop a contextual multi-armed bandit (CMAB) based solution by leveraging techniques from reinforcement learning. The key design philosophy for selecting the CMAB is that it can embed the *context information* (e.g., the number of social media users and their reported events) into the mechanism of the source selection. Incorporating the context information in our model is important since it provides an abstraction of the dynamic states of the social signals and guidance for the TSS module to choose the threshold that is optimal at the current context.

In CMAB, a set of actions are available with each of them associated with a reward. The key idea of CMAB is to observe a context and identify a sequence of actions that can maximize an expected reward with limited or no initial knowledge about each action's reward function. In particular, our problem involves an *exploitation-exploration* tradeoff: the exploration propels the system to try out different actions for acquiring knowledge about the reward functions at the cost of receiving low rewards (i.e., similar to playing slot machines at random in a casino). The exploitation tries to greedily select the action that maximizes the estimated reward. However, the pure exploitation may lead to a poor system performance due to the lack of knowledge of the reward functions without exploration. Our goal is to achieve a good balance between leveraging existing knowledge (i.e., *exploitation*) and attempting new source selection thresholds (i.e., *exploration*).

We now define the key terms in CMAB and its mapping to our source selection problem below:

DEFINITION 10. Context z_l: The context z_l in the TSS represents an abstraction of the dynamic states of the social signals. In particular, we represent the context by a tuple $z_l = \{M_l, \nu_l\}$, where M_l is the number of sources at sensing cycle l and ν_l is the average number of tweets made by a source at sensing cycle l. Intuitively, with greater values of M_l and ν_l, the system will be in a context of more liberty in discarding the unreliable sources and vice versa.

DEFINITION 11. Action Set $\mathcal{A}^l$: The action in the TSS module refers to a set of threshold levels for unreliable sources to be discarded at a particular sensing cycle. It is represented by a set of discrete actions $\mathcal{A}^l = \{A_1^l, A_2^l, ..., A_n^l\}$. An example action set is $\{0.0, 0.1, 0.2, ..., 0.9\}$. A selected action of 0.2 indicates the user U_u will be discarded if $\mu_u \leq 0.2$.

DEFINITION 12. Reward $\mathcal{R}^l$: The reward in the TSS module refers to the accuracy of events identified by the RSSD module. Thus, higher the accuracy, higher the reward. In particular, the reward for sensing cycle l is computed by:

$$\mathcal{R}^l = \frac{1}{N_l}\sum_{p=1}^{N_l}(1 - |\widehat{E_{l,p}} - E_{l,p}|) \tag{9}$$

where $\widehat{E_{l,p}}$ denotes the estimated truth by the RSSD module and $E_{l,p}$ denotes the actual ground truth of the events verified by the drones.

In each sensing cycle l, the algorithm chooses an action $a_l \in \mathcal{A}^l$, and receives reward $\mathcal{R}^l|a_l$ whose expectation depends on both the context z_l the action a_l. We define the total *actual* reward as $\sum_{l=1}^{L}\mathcal{R}^l|a_l$ and the total *optimal expected* reward as $\mathbf{E}[\sum_{l=1}^{L}\mathcal{R}^l|a_l^*]$, where a_l^* is the action with the maximum expected reward at sensing cycle l. The goal of CMAB is to derive an optimal source selection threshold that minimizes the *regret* which is the difference between the actual reward and the optimal reward. Formally, our objective is:

$$\underset{\mathcal{A}^l}{\text{argmin}}\left(\sum_{l=1}^{L}\mathcal{R}^l|a_l - \mathbf{E}\left[\sum_{l=1}^{L}\mathcal{R}^l|a_l^*\right]\right), 1 \leq l \leq L \tag{10}$$

To find the optimal action, we adopt the LinUCB algorithm [37] to solve our CMAB problem. The key mechanism of the LinUCB algorithm is to obtain the expected reward of each action by finding a combination of the past rewards of the action. In particular, we first design a feature vector $x_{l,a}$ to encode the context z_l and action a. We assume that the system initiates a training phase for i sensing cycles, where i is a tunable parameter whose setting is elaborated in Section V. In the first i sensing cycles, we *explore* all the available actions to obtain a set of training data. We define two features in our model (i.e., the number of sources and the average number of tweets made by a source) and use j to indicate the number of features: $j = 2$ in our case). We consider individual feature sets k_a to contain the data of the features in the past sensing cycles. The goal of the algorithm is to learn the relationship between the context-action and reward to establish a model for deciding the optimal source selection threshold. We let J_a be a regressor matrix of dimension $i \times j$ at sensing cycle l. We apply ridge regression to the training data (J_a, k_a) in order to obtain a coefficient vector $\hat{\theta_a}$ during the exploration phase:

$$\hat{\theta_a} = (J_a^T J_a + I_d)^{-1} J_a^T k_a \tag{11}$$

where I_d is a $d \times d$ identity matrix. For notation convenience, let $Q_a = J_a^T J_a + I_d$. After performing the regression, we can obtain the estimated reward by linear combinations of the optimization parameter $\hat{\theta_a}$ and the feature vector $x_{l,a}$:

$$\hat{\mathcal{R}^l} = x_{l,a}^T \hat{\theta_a} \tag{12}$$

In the exploitation phase, we obtain our best action-selection strategy at each sensing cycle l by choosing the action that yields the largest predicted reward for the currently observed context. The variance $\sigma^2_{\hat{\mathcal{R}^l}}$ of the estimated reward is given by:

$$\sigma^2_{\hat{\mathcal{R}^l}} = x_{l,a}^T Q_a^{-1} x_{l,a} \tag{13}$$

Combining the results above, we derive the optimal action (i.e., source selection threshold) as:

$$\hat{a_l^*} = \underset{a \in \mathcal{A}^l}{\text{argmax}}\left(\hat{\mathcal{R}^l} + \beta\sqrt{x_{l,a}^T Q_a^{-1} x_{l,a}}\right) \tag{14}$$

where β is a constant factor and the term $\sqrt{x_{l,a}^T Q_a^{-1} x_{l,a}}$ denotes the standard deviation of the estimated reward. Once the optimal action is learned for the current context, the system switches to using the learned action $\mathcal{A}^l$ for the TSS module.

V. Evaluation

In this section, we extensively evaluate the performance of SocialDrone through a real-world disaster case study. The results demonstrate that SocialDrone significantly outperforms the baselines by identifying the disaster events more accurately and quickly.

A. Experimental Platform and Setup

We first acknowledge the fact that a real-world deployment of drones in disaster response applications is difficult because the disaster situations are unpredictable and can hardly be replicated. Therefore, we implemented our SocialDrone framework in ArduPilot SITL (Figure 5) [38], a reputed drone simulator which can closely reproduce real life disaster scenarios. The simulator can virtually model any UAV with known physical parameters like dimensions, weight, speed, energy consumption and also environmental elements like wind or signal losses. The simulator's integration with Google Maps enables it to simulate sending drones to real-world locations, which is crucial for our experiment. We utilized DroneKit [39], a middleware to connect the Ardupilot SITL engine with the SocialDrone framework via a low-latency UDP connection. The particular model of drone that we selected in our simulator is the DJI Phantom 2, a quad-rotor helicopter [1]. Figure 5 illustrates a snapshot of the SITL simulator interfaced with the SocialDrone framework.

[1]https://www.dji.com/phantom-2/info#specs

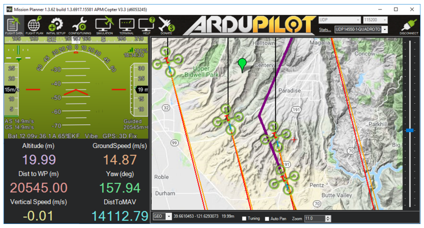

Figure 5. Ardupilot SITL Integrated with SocialDrone. The right pane displays the current position of the drones. The left pane provides the selected drone's altitude, speed, and other critical parameters.

B. Evaluation Dataset

We collected a real world dataset using Twitter data feeds during the California Camp Fire, a wildfire in Northern California that occurred in November 2018. The fire resulted in 86 deaths, 3 missing, 17 injuries including firefighters and damages estimated to be about \$10 billion [2]. The statistics of the dataset are summarized in Table I.

Table I
DATA STATISTICS

Start Date	November 08, 2018
Time Duration	4 days
Location	California, USA
No. of tweets	140,028
No. of tweet users	138,214
No. of event locations	124

The collected data trace is then replayed to emulate the disaster event in real time. We organize the reported events based on their timestamps and group them into a series of sensing cycles. Specifically, we selected the duration of the sensing cycle to be 10 minutes based on the frequency of the events observed in our dataset. There are 500 sensing cycles in the duration of the event we studied. During each sensing cycle, a set of preprocessing steps are performed in real-time. We filter the relevant tweets by first running keyword searches (e.g., Camp, Fire, Road, Missing, California) and remove the irrelevant ones. We also filtered out all tweets without geo-location information. Next, we cluster similar tweets into the same cluster using the state-of-the-art online tweet clustering tool [3] and generate claims that report events at particular locations. We independently collect ground truth labels of the reported events from historical facts and articles published by credible sources (e.g., reports from mainstream news media).

We selected the dimension of each sensing cell to be 1 mile $\times 1$ mile and we have 896 cells in the studied area defined by our dataset. By examining the events in the California Camp Fire, we classify their priorities into three categories based on their urgency, denoted by $\rho_{l,p}$, and assign the event deadlines accordingly. For example, we assigned a deadline range of [1, 3) minutes to the most urgent events ($\rho_{l,p}$=1), a deadline range of [3, 6) minutes to the next level of event urgency ($\rho_{l,p}$=2) and a deadline range of [6, 10] to the last level ($\rho_{l,p}$=3) within a sensing cycle. We then pick a deadline for an event randomly from the above ranges based on its urgency level. The values of the parameters λ_1, λ_2 and λ_3 in the DTA module are obtained through a learning phase. In particular, we select the first $1/5^{th}$ of the sensing cycles as the learning phase. For the TSS module, the value of parameter i is set to 100.

[2]Statistics obtained from https://www.insurancejournal.com/news/west/2018/11/19/509677.html

C. Compared Baselines

We choose the following baselines in our evaluation.

1) Social Only Systems: A social media sensing based system determines the truthfulness of disaster events solely based on the social media sensing data (i.e., tweets). We choose two widely adopted truth discovery algorithms, Hubs and Authorities (HITS) [40] and Maximum Likelihood Estimation (MLE) [5], as our social only schemes.

2) Drone Only Systems: A drone only system utilizes a drone's on-board sensors to perform an aerial scan of a site and identify disaster events. We assume that the drones can obtain the ground truth of the events they covered. We choose two typical patrolling strategies adopted by the autonomous drone guidance systems: i) Random Walk, and ii) Fixed Routes. In Random Walk, drones traverse an arbitrary number of sensing cells in a random direction and changes direction periodically. In Fixed Routes, drones traverse along a designated patrol route in an endless loop and cover events as they come across.

3) Social and Drone Combined Systems: In addition to social only and drone only schemes, we also include several simplified versions of the SocialDrone system as our baselines. We generate the first baseline "MLE+BGT" by keeping the bottom-up game theoretic (BGT) task allocation module and disabling path-planning module and the source selection module in SocialDrone. Next, we take the "MLE+BGT" scheme and further modify the task allocation module to only consider the distance of events from the drones using a *greedy shortest path (GSP)* strategy, calling the scheme "MLE+GSP". We additionally include two more baselines "HITS+GSP" and "HITS+BGT" by replacing the MLE algorithm with HITS in the two aforementioned MLE based baselines.

Note that we assign initial locations of the drones as uniformly distributed across the sensing area for all the baselines with drones. This is because we assume no prior knowledge of where the events could happen and the uniform distribution maximizes the coverage of the drones.

D. Evaluation Results

In the first set of experiments, we evaluate the performance of all schemes across the entire dataset. We selected a set of 15 drones for the testing of our framework as well as the baseline drone based sensing systems. The results are presented in Table II. We observe that SocialDrone consistently outperforms the other approaches in identifying the truthful events during the disaster. In terms of classification accuracy, precision, recall and F1 Score, the performance gain achieved by SocialDrone compared to the best-performing baseline, "MLE+BGT" are 13.7%, 21.2%, 7.5% and 15.6% respectively. Such performance gains of SocialDrone are mainly achieved

Table II
OVERALL PERFORMANCE WITH CALIFORNIA CAMP FIRE DATASET

Category	Algorithm	Accuracy	Precision	Recall	F1-Score
Social Only	HITS	0.443	0.457	0.681	0.547
	MLE	0.648	0.662	0.832	0.738
Drone Only	Rand. Walk	0.272	0.314	0.554	0.400
	Fixed Route	0.427	0.492	0.680	0.571
Social+Drone	HITS+GSP	0.705	0.602	0.816	0.693
	HITS+BGT	0.719	0.615	0.842	0.711
	MLE+GSP	0.751	0.669	0.886	0.762
	MLE+BGT	0.754	0.675	0.883	0.765
Our Scheme	**SocialDrone**	**0.891**	**0.887**	**0.958**	**0.921**

by its closed-loop design that seamlessly integrates the social media sensing and drone system components.

In the second set of experiments, we studied the effect of the number of drones on the performance of the schemes involving drones. For the social and drone combined baselines, we only present the two best performed ones MLE+GSP and MLE+BGT. In the experiment, we vary the number of drones from 10 to 20. Figures 6 report the results for F1 score for the drone based schemes. We observe that the SocialDrone outperforms all the compared schemes when the number of drones changes in the system. The performance gains achieved by SocialDrone also increases as the number of drones in the system increases. The reason is that SocialDrone has more flexibility in terms of both task allocation and path planning when the number of drones increases. The results on precision, recall and accuracy are similar and we do not include them here due to the space limit.

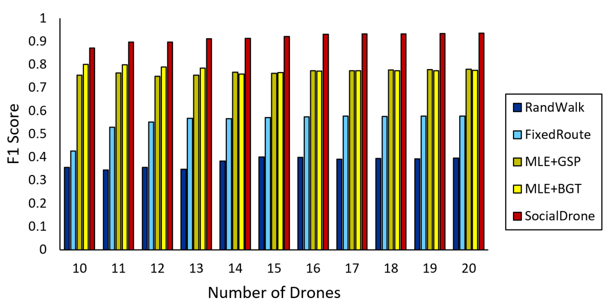

Figure 6. F1 Score vs. Number of Drones for Schemes with Drones

In the third set of experiments, we analyze the deadline hit rate of the drone based schemes in the system. Figure 7 shows the results. We observe that SocialDrone achieves the highest deadline hit rate when the number of drones changes. This is achieved by the deadline aware path planning algorithm design in SocialDrone system. We also observe that the deadline hit rate increases across all schemes as the number of drones increases and it stabilizes when the number of drones becomes large enough (i.e, 16 in our results). This reason is that there are certain events that can never be explored by any of the schemes either because the deadlines are too short or the events are located so far away from the drones that they cannot be explored within their deadlines.

In the last set of experiments, we evaluate the average power consumption of the drone based schemes. The results are shown in Figure 8. We observe that the SocialDrone framework consumes the least amount of power in all the cases regardless of the number of drones. This remarkable savings in power consumption is mainly achieved by the robust path planning algorithm that carefully plans the flight path of the drone by considering drone mechanics and energy consumption. It refrains the drones from flying to events involving infeasible outcomes (e.g., events that have too short a deadline or are too far away and cannot be flown to with successfully meeting the deadline requirement). Predicting ahead of time about the outcomes allows the drones in SocialDrone to take decisive actions ahead of time instead of wasting energy. All the other schemes fall behind as they brute-force exploring all the events. The above results demonstrate that along with recovering truthful state of the events more accurately, SocialDrone also comparatively conserves more energy of drones.

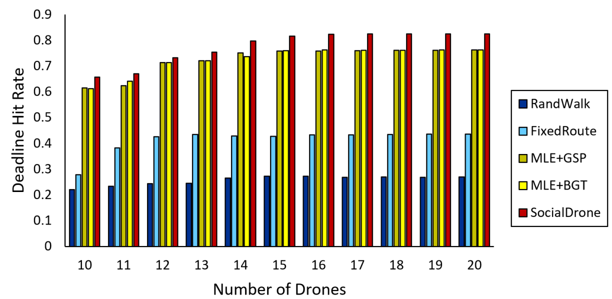

Figure 7. Deadline Hit Rate vs. Number of Drones for Schemes with Drones

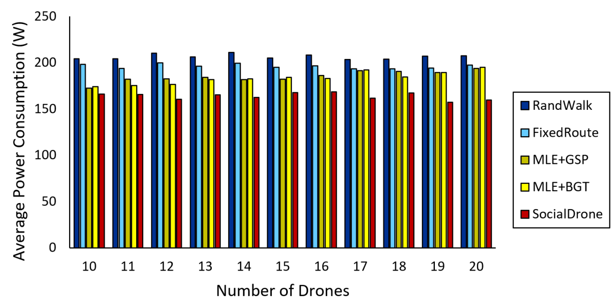

Figure 8. Average Power Consumption (in Watts) vs. Number of Drones for Schemes with Drones

VI. CONCLUSION

In this paper, we develop the SocialDrone framework to integrate social sensing with drone based physical sensing for reliable disaster response. In particular, we develop a reliable social signal distillation module to analyze the event truthfulness and the source reliability. We construct a game theoretic drone task allocation module that leverages the distilled social signals to selectively dispatch drones to the desired locations for active sensing. We further design a closed-loop source selection module that utilizes the drones' sensing measurements to discard unreliable social media users. The results from an extensive evaluation with a real-world dataset show that the SocialDrone scheme significantly outperforms both the social only and drone only systems. The outcomes of this paper motivates a brand new automatic yet effective disaster response system for future disasters.

REFERENCES

[1] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, and L. Kaplan, "The age of social sensing," *arXiv preprint arXiv:1801.09116*, 2018.

[2] T. Baldwin, P. Cook, M. Lui, A. MacKinlay, and L. Wang, "How noisy social media text, how diffrnt social media sources?" in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 2013, pp. 356–364.

[3] D. Y. Zhang, C. Zheng, D. Wang, D. Thain, X. Mu, G. Madey, and C. Huang, "Towards scalable and dynamic social sensing using a distributed computing framework," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017.

[4] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, Jun. 2008.

[5] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. ACM/IEEE 11th Int Information Processing in Sensor Networks (IPSN) Conf*, Apr. 2012, pp. 233–244.

[6] D. Y. Zhang, D. Wang, and Y. Zhang, "Constraint-aware dynamic truth discovery in big data social media sensing," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 57–66.

[7] D. Y. Zhang, R. Han, D. Wang, and C. Huang, "On robust truth discovery in sparse social media sensing," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1076–1081.

[8] X. Yin and W. Tan, "Semi-supervised truth discovery," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011.

[9] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, no. 1, pp. 24–32, 2017.

[10] J. Cohen, "Drone spy plane helps fight california fires," 2007.

[11] A. Restas, "Drone applications for supporting disaster management," *World Journal of Engineering and Technology*, vol. 3, pp. 316–321, 2015.

[12] P. Neumann, M. Bartholmai, J. H. Schiller, B. Wiggerich, and M. Manolov, "Micro-drone for the characterization and self-optimizing search of hazardous gaseous substance sources: A new approach to determine wind speed and direction," in *2010 IEEE International Workshop on Robotic and Sensors Environments*. IEEE, 2010, pp. 1–6.

[13] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Coordination and control of multiple uavs," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2002, p. 4588.

[14] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[15] P.-E. Hladik, H. Cambazard, A.-M. Déplanche, and N. Jussien, "Solving allocation problems of hard real-time systems with dynamic constraint programming," in *14th International Conference on Real-Time and Network Systems (RTNS'06)*, 2006, pp. 214–223.

[16] S. Sariel and T. R. Balch, "Efficient bids on task allocation for multi-robot exploration." in *FLAIRS Conference*, 2006, pp. 116–121.

[17] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.

[18] D. Wang, T. Abdelzaher, and L. Kaplan, *Social sensing: building reliable systems on unreliable data*. Morgan Kaufmann, 2015.

[19] J. R. Harrald, "Agility and discipline: Critical success factors for disaster response," *The annals of the American Academy of political and Social Science*, vol. 604, no. 1, pp. 256–272, 2006.

[20] J. Zander, P. J. Mosterman, T. Padir, Y. Wan, and S. Fu, "Cyber-physical systems can make emergency response smart," *Procedia Engineering*, vol. 107, pp. 312–318, 2015.

[21] Z. Alazawi, S. Altowaijri, R. Mehmood, and M. B. Abdljabar, "Intelligent disaster management system based on cloud-enabled vehicular networks," in *ITS Telecommunications (ITST), 2011 11th International Conference on*. IEEE, 2011, pp. 361–368.

[22] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[23] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "Ccs-ta: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015.

[24] Y. Zhang, D. Zhang, N. Vance, Q. Li, and D. Wang, "A light-weight and quality-aware online adaptive sampling approach for streaming social sensing in cloud computing," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018.

[25] V. Ambati, S. Vogel, and J. G. Carbonell, "Active learning and crowd-sourcing for machine translation." in *LREC*, vol. 1, 2010, p. 2.

[26] L. Sha, R. Rajkumar, and J. P. Lehoczky, "Priority inheritance protocols: An approach to real-time synchronization," *IEEE Transactions on computers*, vol. 39, no. 9, pp. 1175–1185, 1990.

[27] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[28] D. Y. Zhang, Y. Zhang, Q. Li, T. Plummer, and D. Wang, "Crowdlearn: A crowd-ai hybrid system for deep learning-based damage assessment applications," in *Distributed Computing Systems (ICDCS), 2019 IEEE 39th International Conference on*. IEEE, 2019.

[29] S. Ieong, R. McGrew, E. Nudelman, Y. Shoham, and Q. Sun, "Fast and compact: A simple class of congestion games," in *AAAI*, vol. 5, 2005.

[30] H. Ackermann, H. Röglin, and B. Vöcking, "Pure nash equilibria in player-specific and weighted congestion games," in *WINE*. Springer, 2006, pp. 50–61.

[31] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback control theory*. Courier Corporation, 2013.

[32] M. Voorneveld, "Best-response potential games," *Economics letters*, vol. 66, no. 3, pp. 289–295, 2000.

[33] Z. Liu, J. Dai, B. Wu, and H. Lin, "Communication-aware motion planning for multi-agent systems from signal temporal logic specifications," in *American Control Conference, 2017*. IEEE, 2017, pp. 2516–2521.

[34] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3. IEEE, 2002, pp. 1936–1941.

[35] Z. Liu, B. Wu, J. Dai, and H. Lin, "Distributed communication-aware motion planning for multi-agent systems from stl and spatel specifications," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 4452–4457.

[36] Z. Gu, E. Rothberg, and R. Bixby, "Gurobi optimizer reference manual," *URL: http://www. gurobi. com*, vol. 2, pp. 1–3, 2012.

[37] Y. Mandai and T. Kaneko, "Linucb applied to monte carlo tree search," *Theoretical Computer Science*, vol. 644, pp. 114–126, 2016.

[38] I. A. Mason, V. Nigam, C. Talcott, and A. Brito, "A framework for analyzing adaptive autonomous aerial vehicles," in *International Conference on Software Engineering and Formal Methods*. Springer, 2017, pp. 406–422.

[39] "Welcome to dronekit-python's documentation!¶." [Online]. Available: http://python.dronekit.io/

[40] H. Deng, M. R. Lyu, and I. King, "A generalized co-hits algorithm and its application to bipartite graphs," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 239–248.