# Emergency Response Management Pipelines for Smart Cities

Geoffrey Pettet
Vanderbilt University
Nashville, Tennessee, USA
geoffrey.a.pettet@vanderbilt.edu

Ayan Mukhopadhyay
Stanford University
Palo Alto, California, USA
ayanmukh@stanford.edu

Sayyed Mohsen Vazirizade
Vanderbilt University
Nashville, Tennessee, USA
s.m.vazirizade@vandebilt.edu

Matthew Berger
Vanderbilt University
Nashville, Tennessee, USA
matthew.berger@vanderbilt.edu

Mykel Kochenderfer
Stanford University
Palo Alto, California, USA
mykel@stanford.edu

Abhishek Dubey
Vanderbilt University
Nashville, Tennessee, USA
abhishek.dubey@vanderbilt.edu

## ABSTRACT

Emergency response is one of the most pressing problems faced by communities across the globe. It is also one of the most important engagements of governments. In the last fifty years, developing statistical, analytical and algorithmic approaches to design emergency response management (ERM) systems has garnered a lot of attention. An ERM pipeline is an intricate combination of several modular sub-components that need to be individually understood in order to make the overall system work. We have identified several key challenges and worked on different components of the ERM pipeline. A major hurdle for widespread adoption of state-of-the-art ERM techniques is the lack of an *integrated*, ready-to-use ERM toolset that allows policy makers to directly compare various techniques and visualize their impact. We are developing such an open-source toolset, the first of its kind, which integrates four major components of ERM – incident forecasting, strategic (long term) resource allocation, dynamic resource allocation, and dispatch – in a ready to use package. In this paper, we highlight our work on alleviating some of the major challenges of that ERM faces today, and talk about how such approaches are integrated in our tool.

## KEYWORDS

Emergency Response Management, Spatial-Temporal Forecasting, Resource Optimization, Visualization Tools

**ACM Reference Format:**
Geoffrey Pettet, Ayan Mukhopadhyay, Sayyed Mohsen Vazirizade, Matthew Berger, Mykel Kochenderfer, and Abhishek Dubey. 2020. Emergency Response Management Pipelines for Smart Cities. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

Emergency response management (ERM) is a critical problem faced by communities across the globe. First-responders are constrained by limited resources, and must attend to different types of incidents like traffic accidents, fires, crime, and distress calls. Incident response is further complicated by the constraint that quick and timely service is essential. Indeed, it has been noted that the odds of survival of the patient varies non-linearly with response times [1, 2]. As a consequence, statistical and algorithmic approaches to emergency response have received significant attention in the last few decades. Governments in urban areas are increasingly adopting methods that enable *smart* emergency response, which are a combination of forecasting models and visualization tools to understand where and when incidents occur, and optimization approaches to allocate and dispatch responders.

The key components of an ERM pipeline are: (1) *Incident forecasting* – understanding where and when incidents occur , (2) *Long-term resource allocation* – strategic decisions on long-term resource placement, such as how many stations and vehicles to acquire and where to build said stations , (3) *Dynamic resource allocation* – short term operational decisions such as rebalancing vehicle allocations based on current demands , and (4) *Dispatching* – policy for deploying responders when an incident is reported.

While there has been significant research on each individual component of ERM [3, 4], a key limitation of canonical approaches in ERM is that they look at each component separately. The overall functioning of an ERM system is however entirely dependent upon the intricate dependencies of the components. In this paper, we summarize our prior and current work on designing an integrated emergency response pipeline. Specifically, we explain our work on online incident forecasting and algorithmic approaches to decentralized multi-agent response. We highlight how the individual components improve upon the state-of-the-art, and also discuss how they are integrated. Finally, we highlight our efforts on creating an open-source integrated emergency response tool, the first of its kind, that enables first-responder organizations to access the state-of-the-art algorithmic and modeling advancements in emergency response.

## 2 PROBLEM SETUP

Our goal is to develop a generic ERM pipeline that can be used by policy makers across the globe. ERM is typically a major problem in urban communities, which are examples of dynamic, continuous-time, and stochastic environments. We make several assumptions on the problem structure and information provided *a-priori*. First, we assume that the spatial area under consideration is segmented


*Conference'17, July 2017, Washington, DC, USA*

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

into a set of discretized units $G$. This discretization can either consist of a grid of equally sized cells, or a graph of roadway segments. Second, we assume that for each spatial unit, the temporal distribution of incidents is homogeneous. Our third assumption is that emergency responders can wait in a fixed collection of depots $D$, a subset of cells that are analogous to fire-stations. Each depot $d \in D$ has a fixed capacity $C(d)$ of responders that it can accommodate at a time. We assume that when an incident happens, a free responder (if available) is dispatched to the site of the incident. Once dispatched, the time to service consists of two parts – 1) time taken to travel to the scene of the incident, and 2) time taken to attend to the incident. If no free responders are available, then the incident enters a waiting queue.

We consider two major parts of the ERM model in this paper. First, we assume that the spatial-temporal occurrence of the incidents can be represented by the function $f(x \mid w)$, where $x$ is a random variable that represents a measure of incident occurrence (count of incidents or time between incidents for example), and $w \in \mathbb{R}^m$ is a set of spatio-temporal features. Historically, $f$ has been modeled by both probabilistic models (a Poisson distribution, for example), or regression approaches without direct probabilistic interpretations (linear regression, for example). Second, we consider that the control problem involved in an ERM pipeline can be modeled as a Multi-Agent Semi-Markov Decision Process (M-SMDP) [5, 6, 7], which can be represented by the tuple An SMDP system can be described by the tuple $(\Lambda, S, \mathcal{A}, P, T, \rho(i,a), \mathcal{T})$, where $\Lambda$ is a finite collection of agents and $\lambda_j \in \Lambda$ denotes the $j^{\text{th}}$ agent. The action space of the $j^{\text{th}}$ agent is represented by $A_j$, and $\mathcal{A} = \prod_{i=1}^{m} A_j$ represents the joint action space. We assume that the agents are cooperative and work to maximize the overall utility of the system. $S$ is a finite state space of the problem, $P$ is the state transition function and $T$ denotes the temporal transition and $\rho$ represents the reward function. Transitions in ERM depend on several random variables, including incident arrival, travel times of responders, and service times, and cannot be typically modeled in closed-form expressions [5].

A major problem with principled dispatching approaches is that they evaluate decisions *post-incident*. This explains why first responders rarely use such approaches in practice, since in situations where time is of the essence, moral constraints dictate that the closest responder be dispatched to the scene of the incident. To this end, we propose dynamically rebalancing the spatial distribution of responders *between* incidents. We explain how the M-SMDP formulation accommodates this in section 4.

The overall goal of the pipeline is therefore two-fold - first, given a set of historical incidents and associated spatio-temporal features, the decision-maker seeks to learn $f(x \mid w)$, and secondly, she seeks to find an optimal policy for the stochastic control problem represented by the SMDP.

## 3 INCIDENT FORECASTING

### 3.1 Data Aggregation and Clustering

A challenge in learning incident prediction models is data sparsity. An arbitrary spatial unit is unlikely to experience any incident on any particular time-period. The sparsity, combined with high data dimensionality makes model inference difficult. Learning one model for $G$ ignores similarities in incident occurrence that are not explicitly captured by the feature space, while learning one model per member of $G$ results in over-fitting.

A technique to address this is to combine similar spatial units into groups using clustering techniques, with the hypothesis that similar units will have similar incident distributions. Incident models can then be trained on each group rather than on each unit. To this end, we categorize the feature set $w$ into *static* and *dynamic* features [8]. Static features capture covariates that do not change (or stay relatively similar) with time, while dynamic features change over time. Static features for traffic accident analysis can include the curvature of the road, roadway type (e.g. highway, major surface road, or residential), the speed limit, and so on. Aggregated dynamic features can also be used as a static feature, such as mean traffic observed on the roadway. The set of static features can be used to cluster spatial units into groups of *similar* units. Once spatial units are grouped, dynamic features are used to learn a forecasting model over incident arrival. Dynamic features include weather, traffic, and past incidents occurring in the spatial unit or adjacent units.

The accuracy of the learned models depends on the specific grouping of units used, making their construction important. Clustering is a highly interactive analytic task; often, it has several metrics for determining what grouping is *best*. We propose an interactive visual analytics tool to help analysts compare different clustering techniques and their hyperparameter choices. There are two goals of this tool. First, we seek to integrate clusters with a geo-spatial data view to help analysts understand how clusters are distributed on the roadway network, which is important for understanding their underlying characteristics. Secondly, we integrate the cluster visualization and incident forecasting models into one cohesive tool.

Figure 1 shows the *method selection* and *map* views from a prototype of this tool. Users are able to select from several different clustering runs using different clustering techniques and hyperparameters. The tool currently supports $k$-means, DBSCAN, spectral clustering, and hierarchical clustering techniques. Users can quickly compare key clustering metrics for each run, including their silhouette coefficients [9], Calinski-Harabasz indices [10], and Davies-Bouldin indices [11]. Once a run is selected, users can directly see the cluster assignments for each segments and their geo-spatial distribution.

Another challenge that affects building and deployment of ERM pipelines is the extraction and processing of the myriad features that affect incident occurrence, which is complex and time-consuming. We integrate feature processing directly into our tool by enabling end-users to connect with weather data API-s, and by providing approaches to process features. Such capability is important, since data analysis and software development skills needed for feature extraction cannot be expected of all ERM policy makers. We think that it is crucial that tools designed for emergency response enable choosing and extracting features of interest; without such capability, tools are often times not used to their fullest potential, thus increasing the gap between theory and practice in this field.

To illustrate an example of complex feature extraction, we demonstrate how we compute curvature of roadway segments. Curvature represents the change in angle when an object traverses the curve, and is known to be an important predictor of accident rates [12, 13].

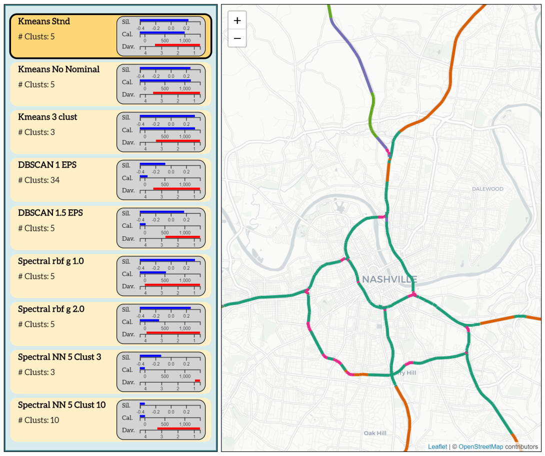



**Figure 1: Visual Tool for Clustering Analysis.**

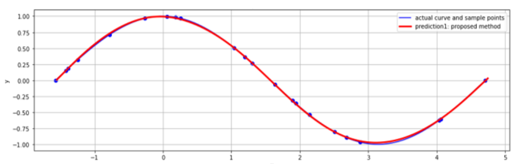



**Figure 2: A Schematic view of a curve and its imitation**

Despite this, accident prediction methodologies in the field often do not use curvature as a feature, since it is lacks an off-the-shelf data source. In practice, one lacks explicit mathematical formulations of roadways, and curvature must be estimated using the available sample points.

To estimate curvature, we fit a spline through the sample points, which provides an explicit function for the geometry of the roadway. Consequently, the curvature at any point on the roadway can be estimated. Figure 2 shows a sample sinusoidal wave (represented by the blue line), and a set of random sample points on it (shown by the blue dots). It also shows the predicted sinusoidal wave (the red line) by employing the proposed method.

## 3.2 Modeling

Formally, we want to learn a probability distribution over incident arrival in space and time. Our tool supports multiple forecasting models, but we leverage our prior work with survival modeling to explain how spatio-temporal models of incident occurrence are learned. Survival models have proven to be extremely effective in predicting incidents like crimes and traffic accidents [14, 8, 5]. A parametric survival model over incident arrival can be represented as $f(\tau|\gamma(w))$. In such a formulation, $f$ is the probability distribution for a continuous random variable $\tau$ representing the inter-arrival time, which typically depends on covariates $w$ via the link function $\gamma$. The link function is typically logarithmic. The optimal model parameters $\beta^*$ can be estimated by the principled procedure of Maximum Likelihood Estimation (MLE).

A limitation of such an approach is that it is *offline*. However, it is imperative to capture the latest trends in incident arrival to accurately predict future incidents, which motivates us to design an online approach for learning and predicting incidents. Consider a set of incidents $D$ used to learn the model parameters $\beta^*$. Now,

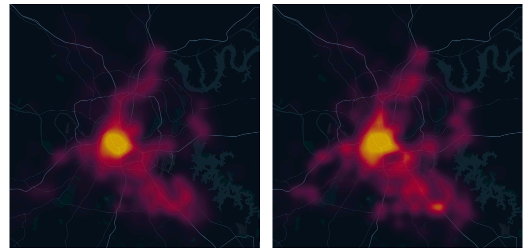


**Figure 3: Heatmaps comparing average incident rates for the forecasting model (left) with actual incidents in Nashville, TN (right)**

.

consider that a set of incidents $D^{'} = \{(x_1^{'}, w_1), (x_2^{'}, w_2), .., (x_k^{'}, w_k)\}$ is available, which consists of incidents that have happened *after* (in time) the original set of incidents. We aim to update the regression coefficients $\beta$ using $D^{'}$, assuming that the model already has access to $\beta^*$.

In order to address this problem, we use stochastic gradient descent to update the distribution $f$ in an online fashion [15]. Formally, we start with the known coefficients $\beta^*$ and, at any iteration $p$ of the process, we use the following update rule

$$\beta^{p+1} = \beta^p + \alpha \nabla L(\beta^p, D^{'})$$

where $\nabla(L(\beta^*, D^{'})$ is the gradient of the log-likelihood function calculated using $D^{'}$ at $\beta^p$ and $\alpha$ is the standard step-size parameter for gradient based algorithms. The regression parameter is updated iteratively based on a predefined convergence criteria.

Figure 3 visualizes the performance of such a survival model by comparing a heatmap of mean predicted incident rates with the actual incident distribution of Nashville, TN between 1-1-2018 and 1-1-2019. This model used features such as weather and previous nearby incidents (to model incident cascades).

# 4 DYNAMIC RESOURCE ALLOCATION

Dynamic resource allocation is a complex problem which requires solutions that can cope with extremely large state spaces. One possible approach is to directly solve the SMDP model by estimating the transition function [5]. Unfortunately this approach is too slow for dynamically rebalancing the distribution of responders, which for an average-sized metropolitan area, has a cardinality of $10^{25}$ [6].

We examine the Monte-Carlo Tree Search (MCTS) family of algorithms, which evaluate actions by sampling from a large number of possible scenarios. A standard MCTS-based approach is not suitable for dynamic allocation due to the sheer size of the state-space in consideration coupled with the low latency that ERM systems can afford. Therefore, we focus on a decentralized multi-agent MCTS (MMCTS) approach explored by Claes et. al [16] for multi-robot task allocation during warehouse commissioning. In MMCTS, individual agents build separate trees focused on their own actions, rather than having one monolithic, centralized tree, dramatically reducing their search space. To realize MMCTS for an ERM domain, some extensions need to be made to standard UCT [17]. Agents must have an accurate yet computationally cheap model of other

| Identifier | Description | Hyper-Parameter Choices |
|---|---|---|
| BASE | Greedy Baseline Without Rebalancing | N/A |
| M-1 | MMCTS - Baseline<br>The foundation for the parameter search. Each parameter varies independently while other parameters retain these values.<br>(All M-* experiments use generated incident chains and a Static Agent Policy) | MCTS Iteration Limit = 250<br>Lookahead Horizon = 120 min<br>Reward Distance Weight $\psi$ = 10<br>Reward Discount Factor = 0.99995<br>Rebalance Period = 60 min |
| M-2 | MMCTS - Iteration Limit of 100 | MCTS Iteration Limit = 100* |
| M-3 | MMCTS - Iteration Limit of 500 | MCTS Iteration Limit = 500* |
| M-4 | MMCTS - Reward Distance Weight $\psi$ of 0 | Reward Distance Weight $\psi$ = 0* |
| M-5 | MMCTS - Reward Distance Weight $\psi$ of 100 | Reward Distance Weight $\psi$ = 100* |
| M-6 | MMCTS - Rebalance Period of 30 minutes; Lookahead Horizon of 30 minutes | Lookahead Horizon = 30 min<br>Rebalance Period = 30min* |

**Table 1: Outline of the experimental runs performed and their corresponding hyper-parameter choices. (*When not indicated, parameters are set to values of M-1, the MMCTS Baseline in the table.)**

agents' behavior. To this end, we use an queue-based rebalancing heuristic (described in [6]) to approximate agent behavior.

Another extension we make to standard MCTS approaches is *action filtering*. The dispatching domain has several global constraints to adhere to, such as ensuring that an incident is serviced if possible. We propose a filtering step be added to the MMCTS workflow. Once each individual agent has scored and ranked each possible action, these are sent to a centralized filter that chooses the final actions for each agent to maximize utility without breaking any constraints.

The evaluation function is split into cases reflecting the separate *incident dispatch* and *balancing* steps in our solution approach. For a state $s$ in the tree of agent $\lambda_j$, we design the reward $\rho$ of taking an action $a$ in $s$ as

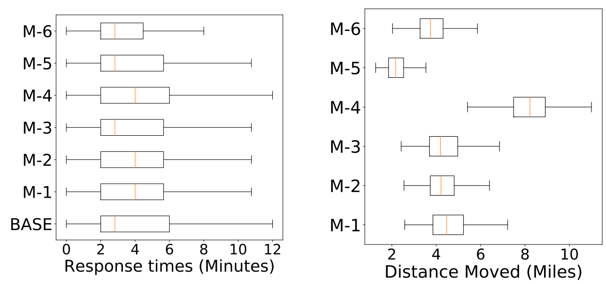
$$\rho(s,a) = \begin{cases} \rho_{s-1} - \alpha^{t_h}(t_r(s,a)), & \text{if responding to an incident} \\ \rho_{s-1} - \alpha^{t_h}\psi\frac{\sum_{\lambda_k \in \Lambda}(\phi_k(s,a))}{|\Lambda|}, & \text{if balancing at } s \end{cases} \tag{1a}$$

where $\rho_{s-1}$ refers to the total accumulated reward at the parent of state $s$ in the tree, $\alpha$ is the discount factor for future rewards, and $t_h$ the time since the beginning of the planning horizon $t_0$. In a dispatch step, the reward is updated with the discounted response time to the incident $t_r(s,a)$. In a balancing step, we update the reward by the average distance traveled by the agents (we denote the distance traveled by agent $\lambda_k$ while balancing due to action $a$ in $s$ by $\phi_k(s,a)$). $\psi$ is an exogenous parameter that balances the trade-off between response time and distance traveled for balancing, and is set by the user depending on their priorities. Distance is not included during dispatch actions, as we always send the closest agent.

We evaluated the proposed MMCTS approach on data from Nashville, TN. The incident prediction model was trained on 35858 incidents occurring between 1-1-2018 and 1-1-2019, and we evaluated the decision processes on 2728 incidents occurring in the month of January, 2019. The parameters for each experimental run are described in table 1, and the corresponding results shown in figures 4a and 4b. We observe that with proper parameter choices, the response time variance is reduced without incurring large rebalancing costs (i.e. distance agents move during rebalancing).

# 5 INTEGRATED PIPELINE AND TOOL

We briefly describe the framework and architecture of our open-source response tool (Fig. 5). The tool accesses historical incident




(a) (b)

**Figure 4: (a) Response time distributions for parameter search experiments compared to greedy (base) strategy. (b) Mean miles traveled per agent in each balancing step.**

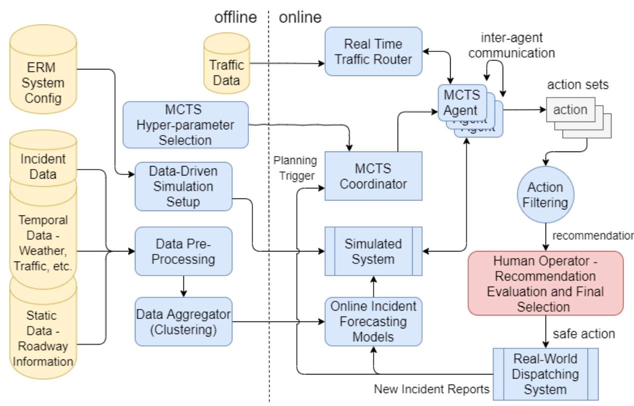



**Figure 5: Response Tool Framework**

data, temporal data (weather and traffic) and static roadway data. The data is processed and fed into an aggregator, which identifies clusters of roadway segments (or any user-specified unit of spatial discretization) that are similar to each other using a visual analytic tool. For each such cluster, a set of online forecasting models are learned and compared automatically with criteria such as test-set likelihood and AIC (Akaike Information Criteria), and are updated as new incidents are reported. Our tool currently supports Poisson regression, negative-binomial regression, parametric survival modeling and zero-inflated Poisson regression. The forecasting model that best describes the data is used by an MMCTS allocation model (under development) which recommends allocation and dispatching actions to human operators. These operators make the final ERM decisions to avoid moral or legal constraints.

# 6 CONCLUSION

Although ERM poses one of the biggest challenges to urban areas across the globe, there are several important gaps between theory and practice in this field. We have systematically bridged several of these gaps by creating online models for spatial temporal forecasting and designing fast and tractable solutions to stochastic control problems that can aid dispatch and allocation of responders. We are combining the entire set of approaches in an open-source ERM tool, the first of its kind, that will make it possible for policy makers to access the state-of-the-art approaches from research related to emergency management.

## REFERENCES


[1] Henrik Jaldell. "How important is the time factor? Saving lives using fire and rescue services". In: *Fire technology* 53.2 (2017), pp. 695–708.

[2] Henrik Jaldell, Prachaksvich Lebnak, and Anurak Amornpetchsathaporn. "Time is money, but how much? The monetary value of response time for Thai ambulance emergency services". In: *Value in health* 17.5 (2014), pp. 555–560.

[3] B Nambuusi, Tom Brijs, and Elke Hermans. "A review of accident prediction models for road intersections". In: (Apr. 2014).

[4] Dominique Lord and Fred Mannering. "The statistical analysis of crash-frequency data: A review and assessment of methodological alternatives". In: *Transportation Research Part A: Policy and Practice* 44.5 (June 2010), pp. 291–305.

[5] Ayan Mukhopadhyay, Zilin Wang, and Yevgeniy Vorobeychik. "A Decision Theoretic Framework for Emergency Responder Dispatch". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018.* 2018, pp. 588–596.

[6] Geoffrey Pettet et al. "On algorithmic decision procedures in emergency response systems in smart and connected communities". In: *arXiv preprint arXiv:2001.07362* (2020).

[7] Khashayar Rohanimanesh and Sridhar Mahadevan. "Learning to take concurrent actions". In: *Advances in neural information processing systems.* 2003, pp. 1651–1658.

[8] Ayan Mukhopadhyay et al. "Prioritized Allocation of Emergency Responders based on a Continuous-Time Incident Prediction Model". In: *International Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents and Multiagent Systems. 2017, pp. 168–177.

[9] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[10] Tadeusz Caliński and Jerzy Harabasz. "A dendrite method for cluster analysis". In: *Communications in Statistics-theory and Methods* 3.1 (1974), pp. 1–27.

[11] David L Davies and Donald W Bouldin. "A cluster separation measure". In: *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), pp. 224–227.

[12] Mark Poch and Fred Mannering. "Negative binomial analysis of intersection-accident frequencies". In: *Journal of Transportation Engineering* 122.2 (1996), pp. 105–113.

[13] S. Hadi Khazraee, Valen Johnson, and Dominique Lord. "Bayesian Poisson hierarchical models for crash data analysis: Investigating the impact of model choice on site-specific predictions". In: *Accident Analysis & Prevention* 117 (Aug. 2018), pp. 181–195.

[14] Ayan Mukhopadhyay et al. "Optimal Allocation of Police Patrol Resources Using a Continuous-Time Crime Model". In: *Conference on Decision and Game Theory for Security.* 2016.

[15] Ayan Mukhopadhyay et al. "An Online Decision-theoretic Pipeline for Responder Dispatch". In: *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems.* ICCPS '19. Montreal, Quebec, Canada: ACM, 2019, pp. 185–196.

[16] Daniel Claes et al. "Decentralised online planning for multi-robot warehouse commissioning". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems.* International Foundation for Autonomous Agents and Multiagent Systems. 2017, pp. 492–500.

[17] Johannes Fürnkranz and Tobias Scheffer. *Machine Learning: ECML 2006: 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings.* Vol. 4212. Springer Science & Business Media, 2006.