MobiRescue: Reinforcement Learning based Rescue Team Dispatching in a Flooding Disaster

Li Yan*, Shohaib Mahmud*, Haiying Shen*, Natasha Zhang Foutz† and Joshua Anton‡
*Department of Computer Science, University of Virginia, USA
†McIntire School of Commerce, University of Virginia, USA
‡X-Mode Social, Inc, USA
Email: {ly4ss, sm6re, hs6ms, ynf8a}@virginia.edu, josh@xmodesocial.com

***Abstract*—The effectiveness of dispatching rescue teams under a flooding disaster is crucial. However, previous emergency vehicle dispatching methods cannot handle flooding disaster situations, and previous rescue team dispatching methods cannot accurately estimate the positions of potential rescue requests or dispatch the rescue teams according to the real-time distribution of rescue requests. In this paper, we propose *MobiRescue*, a human *Mobi*lity based *Rescue* team dispatching system, that aims to maximize the total number of fulfilled rescue requests, minimize the rescue teams' driving delay to the rescue requests' positions and also the number of dispatched rescue teams. We studied a city-scale human mobility dataset for the Hurricane Florence, and found that the disaster impact severities are quite different in different regions, and people's movement was significantly affected by the disaster, which means that the rescue teams' driving routes should be adaptively adjusted. Then, we propose a Support Vector Machine (SVM) based method to predict the distribution of potential rescue requests on each road segment. Based on the predicted distribution, we develop a Reinforcement Learning (RL) based rescue team dispatching method to achieve the aforementioned goals. Our trace-driven experiments demonstrate the superior performance of *MobiRescue* over other comparison methods.**

I. Introduction

Flooding disasters (e.g., Hurricane Florence (September 12-15, 2018) and Hurricane Michael (October 7-16, 2018)) occur around the world almost every year and kill much population and cause injuries. Dispatching rescue teams (usually carried by vehicles) to fulfill the pick-up requests of the people that need rescue and deliver them to a safe hospital is crucial for decreasing the number of deaths under flooding disasters. Recent studies [1], [2] have verified that the number of deaths generally increases significantly during and after disaster if the affected population are not rescued timely. Therefore, efficient and effective dispatching of rescue teams under a flooding disaster is important. In this paper, we focus on rescue teams that use vehicles as people carriers.

Many traditional methods that focus on determining the driving routes of emergency vehicles to reduce the vehicles' driving delay to the target emergency event positions under normal situations have been proposed [3]–[6]. The driving delay of a rescue team or a vehicle is defined as the time from its current position to the end of its dispatched destination. Generally, these methods use different models (e.g., integer programming, Markov Decision Process) to consider multiple factors (e.g., traffic status, vehicle availability) in determining the emergency vehicles' driving routes to maximally cover the emergency requests and minimize the vehicles' driving delay in serving the requests. However, these methods aim to serve appearing emergency requests on demand under normal situations but are not applicable for flooding disaster situations since they cannot predict the appearance of rescue requests and proactively guide the rescue teams to timely serve the requests. Long time in solving the integer programming problem further prevents the timely rescue serving.

Generally, the appearance of people's rescue requests is closely related with their mobility in a disaster. To timely serve the rescue requests, the rescue team dispatching center needs to consider the movement of the people that are likely to generate rescue requests when dispatching rescue teams. Several methods that focus on dispatching rescue teams to pick up the people under disaster (e.g., flooding, earthquake) also have been proposed [1], [2], [7]–[9]. These methods apply different stochastic models (e.g., time series analysis) on the distribution of historical rescue request appearances to estimate the future appearance of people's rescue requests, and formulate and solve integer programming problems to guide the rescue teams to a certain destination road segments to rescue the people trapped in disaster with the minimum driving delays. However, these methods do not consider the factors (e.g., precipitation, wind speed, altitude) that reflect the danger level of people's surrounding environment, and thus may have insufficient accuracy in estimating the positions of potential rescue requests since some factors of a position may change in different disasters and even dynamically change in one disaster. The people staying in a more dangerous environment should have a higher priority to be rescued. For example, in a flooding disaster, compared with people living in high-altitude places, the people that are trapped in low-altitude places should be more immediately rescued since the flooding is more likely to endanger their basic living; and people living in an area with strong winds are more likely to need rescue. Also, though time is critical for rescuing people in a flooding disaster, the integer programming problem based dispatching methods are time-consuming and cannot efficiently guide rescue teams in real time to serve rescue requests. Therefore, it is challenging to more accurately predict the distribution of potential rescue requests (i.e., the number of persons that need to be rescued on each road segment) in a flooding disaster, and optimize the dispatching of the rescue teams in real time to maximize the total number of rescued people with the minimum driving delay and minimum number of serving rescue teams.

To handle the challenge, we propose *MobiRescue*, a human *Mobi*lity based *Rescue* team dispatching system that aims to maximize the total number of rescued people, and meanwhile minimize the rescue teams' driving delay to the rescued people's pick-up positions and minimize the number of serving rescue teams. First, we analyzed a city-scale mobility dataset that record the movements of most of the people in the North Carolina State during and after the Hurricane Florence (Sep. 12-15, 2018) and gained the following observations:

(1) We found that the impact severities of the hurricane disaster are quite different in different regions of the city, and can be assessed by the characteristics that describe the disaster (e.g., precipitation, wind speed and altitude). Thus, we can utilize the characteristics to predict the distribution of people's potential rescue requests, which serves as the guidance for dispatching rescue teams.
(2) We also found that people's movement was significantly affected by the disaster. To efficiently fulfill the people's pick-up rescue request, the rescue team dispatching system must be able to adjust the rescue teams' driving routes in real time to adapt to the real-time changes of the distribution of people's potential rescue requests.

The observations serve as the foundation for the design of *MobiRescue* which runs periodically (e.g., every 5 minutes). Accordingly, we first develop a Support Vector Machine (SVM) [10] based method to take into account various disaster-related factors (i.e., characteristics that reflect the intensity of the disaster, such as precipitation, wind speed, and altitude) in predicting the distribution of potential rescue requests. Then, based on the predicted distribution of potential rescue requests and the operable roads (i.e., not destroyed by the disaster) obtained from external support (e.g., satellite imaging of the disaster area from area National Weather Service [11]), we develop a Reinforcement Learning (RL) based method, which can quickly output the guidance for the rescue teams without taking a long time as in calculating the integer programming problem. The inputs to the RL model include the status of each rescue team and rescue requests (current position, the number of rescue requests on each road segment) and the outputs include the guidance on which road segment each rescue team should drive to. The reward is defined in a way that increases the number of rescue requests served by all rescue teams, reduces the total driving delay that the rescue teams need to drive to their assigned pick-up positions, and reduces the total number of serving rescue teams.

In summary, our contributions include:

1. We analyze on a city-scale human mobility dataset in the Hurricane Florence disaster, which confirms the effect of disaster on vehicle flow rate (i.e., the average number of vehicles per unit time) of each road segment and human movement. The analytical results lay the foundation for the design of *MobiRescue*.
2. We propose the *MobiRescue* rescue team dispatching system that aims to maximize the total number of rescued people, and meanwhile minimize the rescue teams' driving delay to the rescue requests' positions and also the number of serving rescue teams.
3. We have conducted extensive trace-driven experiments to show the effectiveness of *MobiRescue* in terms of the number of rescued people per unit time, the rescue teams' average driving delay to the rescue requests' positions, the prediction accuracy and precision of rescue requests, the number of serving rescue teams, and timeliness of rescuing.

To our knowledge, this paper is the first work for rescue team dispatching that considers various disaster-related factors in predicting the appearance of potential rescue requests and adaptively update the driving route of rescue team in real time under flooding disaster situations. Note that the factors taken into account for rescue request prediction can be manually selected according to different catastrophic situations, so our designed method can be extended to other disasters (e.g., earthquake, blizzard, etc.). Since the data is collected under the context of flooding disasters, we limit the paper writing to flooding catastrophic situation specifically. The remainder of the paper is organized as follows. Section II presents the literature review. Section III presents our dataset analysis results. Section IV presents the detailed design of *MobiRescue*. Section V presents our performance evaluations. Section VI concludes the paper.

II. Related Work

Emergency vehicle dispatching under normal situations. Many works [3]–[6] focus on scheduling the driving route of emergency vehicles to reduce the vehicles' driving delay to the target emergency event positions under normal situations. Schmid *et al.* [3] proposed to consider the emergency vehicles' real-time positions in updating their driving routes to minimize the emergency vehicles' driving delay to the emergency event positions. Maxwell *et al.* [4] proposed an approximate dynamic programming (ADP) model that schedules the driving routes of emergency vehicles to maximize the real-time coverage of potential emergency events while keeping the emergency vehicles' service delay below a certain threshold. Van *et al.* [5] proposed an integer programming to deploy the stand-by positions of emergency vehicles to minimize the emergency vehicles' average driving delay to the emergency event positions. Snyder *et al.* [6] further considered the availability of emergency vehicles to dynamically redeploy the stand-by locations of emergency vehicles to minimize the emergency vehicles' driving delay when some areas do not have available emergency vehicles. However, these methods aim to serve appearing emergency requests on demand under normal situations but are not applicable for the flooding disasters.

Rescue team dispatching under catastrophic situations. Many works [1], [2], [7]–[9] have been proposed to optimize the dispatching of rescue teams by utilizing various stochastic models to estimate the appearance of potential rescue requests under flooding disaster situations. Sun *et al.* [7] proposed a fuzzy rough set theory based model to predict the time-varying change of rescue requests based on historical appearance records of rescue requests. Huang *et al.* [8] formulated an integer programming problem to minimize the sum of the driving delays of the rescue teams. Cavdur *et al.* [1] proposed a two-stage stochastic programming model to minimize the

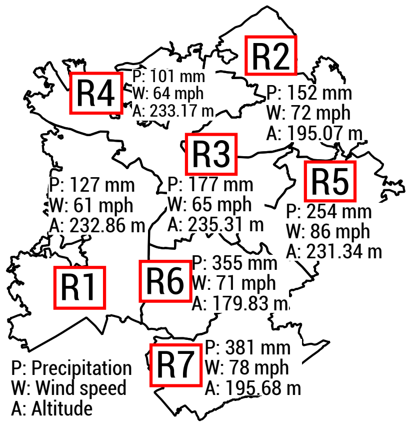


Fig. 1: Regions of Charlotte.

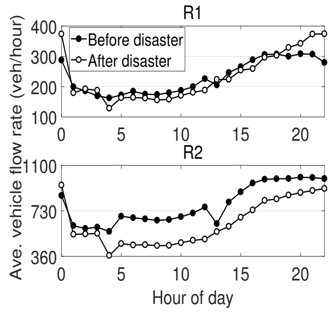


Fig. 2: Average vehicle flow rate of two regions before and after disaster.

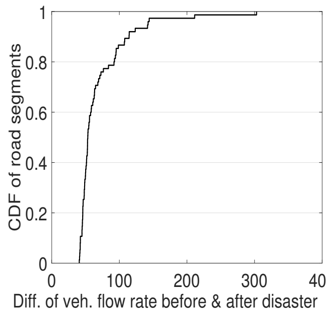


Fig. 3: Difference of average vehicle flow rate before and after disaster.

total travel distance of rescue teams and the number of unmet rescue requests. Edrissi *et al.* [2] studied the reliability measurement of road network under catastrophic situation and proposed a heuristic algorithm to minimize the rescue teams' travel time considering the different reliability of road segments. El-Tawab *et al.* [9] proposed to detect the real-time weather condition of the road network by spreading sensors to roadside after a disaster, and used the sensor data to dispatch the rescue teams with the minimum driving delay to rescue requests. However, these methods have insufficient accuracy in estimating the positions of people's rescue requests as explained previously. Also, their integer programming problem based dispatching methods are time-consuming and cannot efficiently guide the rescue teams in real time to serve the rescue requests. Our *MobiRescue* can handle these problems.

III. Dataset Measurement

A. Dataset Description and Definitions

Our datasets record the human mobility of 8,590 people in the Charlotte city of North Carolina 15 days prior and after the Hurricane Florence (Sep. 12-15, 2018) [12]. They include:

- **GPS Data.** The GPS data is collected from individual's cellphone GPS sensor at a certain time interval (varying from 0.5 to 2 hours) of the 8,590 people. The data contains timestamp, latitude, longitude, altitude, and speed of each user during the sampling period. The associated timestamp and unique ID of each individual allows us to track each user anonymously. The location data covers all over Charlotte.
- **Road map data.** This data of Charlotte is obtained from OpenStreetMap [13]. We have used a bounding box with coordinate (latitude=35.6022, longitude=-79.0735) as south-west corner, and coordinate (latitude=36.0070, longitude=-78.2592) as north-east corner. We have used the data from National Weather Service [11] to crop the affected area.
- **Weather data.** It includes precipitation and wind speed during the disaster and is obtained from National Weather Service.

For data management, we utilized a 11.7 TB Hadoop Distributed File System (HDFS) [14] on a cluster consisting of 52 nodes, each of which is equipped with 20 cores and 32 GB RAM. For data processing, we used Apache Spark [15], which is a fast in-memory cluster computing system running on Hadoop. We study the spatio-temporal correlation and divergence between the mobility (in terms of trips) obtained by cellphones, which serve as the empirical guidelines for our system design later. We represent the road network of Charlotte city with a directed graph $G = (E, V)$, in which vertices V represent landmarks (i.e., intersections or turning points in the road network), and edges E represent road segments [16]–[18]. Based on the road network, we introduce the following definitions:

Definition 1. Trajectory. *A human's or a rescue team's trajectory is a sequence of time-ordered landmarks, where each landmark is represented by a latitude and a longitude.*

Definition 2. Vehicle Flow Rate. *Vehicle flow rate of a road segment is defined as the average number of vehicles driving through the road segment per unit time (an hour) [19]. The vehicle flow rate of a region is defined as the average vehicle flow rate over all the road segments in the region.*

According to the planning of Charlotte City Council Districts, the area of Charlotte City is partitioned into a set of 7 regions, which is shown in Figure 1. In Figure 1, we also put the average precipitation (denoted by P), wind speed (denoted by W) and altitude (denoted by A) per hour during the hurricane (Sep. 14) in each region. The average precipitation and wind speed data is obtained from the National Weather Service. The average altitude is calculated based on the altitude readings of the people's movement data in each region. Since these characteristics reflect the intensity of the hurricane disaster, we define them as *disaster-related factors*.

B. Dataset Analysis

1) Disaster Impact on Different Regions in a City: Generally, the incidents caused by a flooding disaster will severely impact people's movement in the disaster area. To determine whether a person's movement is impacted by flooding, we refer to National Weather Service [11] to obtain the satellite imaging of the flooding zones in the city. We assume that if a person's position is in a flooding zone, the person's movement is severely affected and he/she potentially needs to be rescued to a safer spot. Vehicle flow rate of a road segment is a good indicator on how severe the people's movement is affected [19].

To find whether the hurricane disaster has different impact severities on different regions in the city, we measured the average vehicle flow rate over all the road segments in two regions (R1 and R2 in Figure 1) at each hour on August 25, 2018 (i.e., before disaster) and September 20, 2018 (i.e., after disaster) as our previous foundation work [12]. Figure 2 shows the measured results. We can see that in R1 (P: 127 mm, W: 61 mph, A: 232.86 m), the difference between the average vehicle flow rate before and after the disaster is quite small ($<$ 100 vehiles/hour during all hours). This means that the people's basic living in R1 is not severely affected. They can still utilize the basic road infrastructure in R1 for daily life after disaster. While in R2 (P: 152 mm, W: 72 mph, 195.07 m), the difference between the average vehicle flow rate before and after the disaster is much larger and can be as high as 300 vehiles/hour. This means that the people's basic living in R2 has been impaired by the hurricane disaster. The primary reason is that the average precipitation in R1 (61 mm) is much lower than that in R2 (72 mm), and the average altitude in R1 (232.86 m) is higher than that in R2. Thus, the flooding resulted from the precipitation causes more damage in R2's road infrastructure.

To show the difference of impact severities in different regions, we measured the Cumulative Distribution Function (CDF) of the difference of all road segments' average vehicle flow rate before and after the disaster. The results are illustrated in Figure 3. We can see that most road segments ($> 80\%$) have a difference of average vehicle flow rate higher than 50 vehiles/hour before and after the disaster. The differences of average vehicle flow rate of the road segments vary quite a lot from 50 to 300 vehicles/hour. The above results confirm that the impact severities of the hurricane disaster are quite different in different road segments.

To further confirm that the disaster-related factors can reflect the impact severity of the hurricane disaster, we measured the Pearson correlation coefficient [20] between vehicle flow rate and precipitation, wind speed and altitude, respectively, for all the regions in Charlotte city. The Pearson correlation coefficient is calculated as $\frac{\text{cov}(R,\lambda)}{\sigma_R \sigma_\lambda}$, where R is the vehicle flow rate, λ is one of the three disaster-related factors, cov() is the covariance of R and λ, σ_R and σ_λ are the standard deviations of R and λ. The range of the coefficient is from -1 to +1, where a negative value means the two variables are reversely correlated, a positive values means the two variables are positively correlated. A higher absolute value of correlation coefficient between two variables means that they are more closely correlated. The results are illustrated in Table I. We can see that the result of precipitation is -0.897, the result of wind speed is -0.781, and the result of altitude is 0.739. This means that precipitation and wind speed are reversely correlated with vehicle flow rate, while the altitude is positively correlated with vehicle flow rate. That is, the higher precipitation and wind speed a region has, the lower vehicle flow rate and the region is more severely affected by the disaster, and the higher altitude a region has, the higher vehicle flow rate and the region is less severely affected by the disaster. The absolute values of the correlation coefficients of the disaster-related factors follow: precipitation>wind speed>altitude, which means precipitation has the greatest impact on the change of vehicle flow rate. From the results, we know that the disaster-related factors at a person's position are useful for assessing the impact severity of the position and help more accurately determine whether the person needs rescue. In Section IV-B, we will elaborate the details of our method on combining these factors for predicting the distribution of people's potential rescue requests.

TABLE I: Correlation between disaster-related factors and vehicle flow rate.

	Precipitation	Wind speed	Altitude
Vehicle flow rate	-0.897	-0.781	0.739

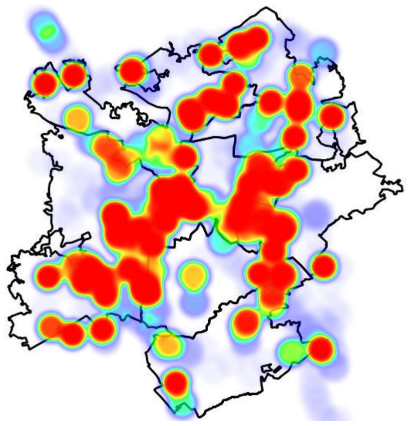


Fig. 4: Region distribution of rescued people.

Observation 1: A disaster has different impact severities on different regions in a city, and disaster-related factors can be used to measure the impact severity of a location.

2) Relationship between Disaster Impact and Rescue Requests: Intuitively, the more greatly the vehicle flow rate (people's movement) is affected, the more people's rescue requests will appear since they are trapped and cannot even drive their vehicles for self-evacuation. To support this conjecture, we analyze the change of vehicle flow rate and the number of people rescued to hospital before, during and after the disaster, and study the relationship between them.

For each region, we first measured the average vehicle flow rate over the 24 hours in each day before the disaster (September 10–13), during the disaster (September 14–16) and after the disaster (September 17–19) as in [12]. Figure 5 shows the results of all the regions. We can see that the average vehicle flow rate during the disaster dropped significantly compared with that before the disaster. During the disaster, the average vehicle flow rates in the 6 regions all dropped to almost 0. After the disaster, although the vehicle flow rates were restored a lot compared to those during the disaster, the results are still much lower than those before the disaster. The gap in the average vehicle flow rates during the days before disaster and the days after disaster can be as high as 300 in Region 3, and around 100 in the other regions. Region 3 has a higher gap compared with the other regions because it is the central downtown area where vehicles frequently drive by when there was no disaster. During the disaster, its vehicle flow rate significantly dropped. This result shows that the people's movement was greatly affected by the disaster.

Second, we measured the total number of people rescued to all the hospitals of Charlotte city. Specifically, we first identified all the people delivered to a hospital in each day. Starting from a person's first appearance in a hospital, if the person stayed at the hospital for more than a time threshold (2

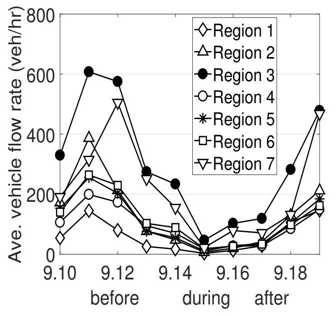


Fig. 5: Vehicle flow rate of each region before, during and after disaster.

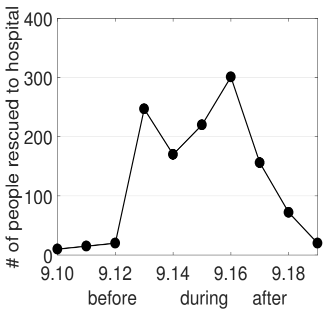
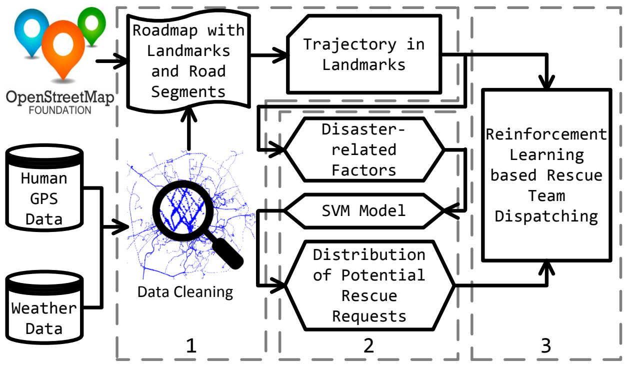


Fig. 6: # of people delivered to hospitals before, during and after disaster.



OpenStreetMap FOUNDATION
Roadmap with Landmarks and Road Segments
Trajectory in Landmarks
Human GPS Data
Weather Data
Data Cleaning
Disaster-related Factors
SVM Model
Distribution of Potential Rescue Requests
Reinforcement Learning based Rescue Team Dispatching
1
2
3

Fig. 7: Framework of MobiRescue.

hours in this paper), we view this person has been delivered to the hospital. However, it is possible that not all people delivered to the hospitals are rescued from flood trapped places. Therefore, we further analyzed each delivered person's previous position before he/she was delivered to the hospital. If the person's previous position is located in a flooding zone according to the satellite imaging explained above, we conclude that this person was trapped by flooding and was rescued to the hospital. Figure 6 shows the final measured results. We can see that starting from September 13 (start of hurricane impact), there is a steep jump in the number of people delivered to hospital. From September 13 to September 16, the number of people delivered to hospital remains very high, which corresponds to the low average vehicle flow rate during the same days in Figure 5. This means that the hurricane flooding disaster greatly impacted all the regions and caused many rescue requests. Figure 4 shows the distribution of the rescued people in different regions. The warmer color (i.e., more red) that a region has, the more rescue requests that appeared in the region, and the colder color (i.e., more green) that a region has, the fewer rescue requests that appeared in the region. We can see that most rescue requests appeared in Region 3, which was impacted the most by the disaster. The results of Figure 5, Figure 6 and Figure 4 confirm that more people's rescue requests appeared under higher disaster impact measured by vehicle flow rate. To adapt to the sudden change of people's rescue requests after disaster, a method that can adjust the driving routes of the rescue teams in real time is necessary. In Section IV-C, we will elaborate the details of our RL-based rescue team dispatching method.

Observation 2: Higher disaster impact leads to more rescue requests. The distribution of people's movement has significant change before, during and after the disaster.

Problem statement: Given the remaining available road network (i.e., road segments that vehicles can still drive through) after diaster provided by satellite imaging (denoted as $\widetilde{G} = (\widetilde{E}, \widetilde{V})$), and real-time distribution of people collected from people's cellphone, how to predict the distribution of people's potential rescue requests and dispatch the rescue teams to maximize the total number of rescued people, minimize the driving delay to the rescue requests' position, and minimize the number of serving rescue teams?

IV. System Design

We assume that each rescue team has one vehicle. The capacity of a rescue team (i.e., the number of persons that can be rescued at a time) is c, which can be set by the rescue team dispatching center according to actual equipment details (e.g., $c = 5$). There will be multiple rounds of dispatches within 24 hours per rescue team. Once a rescue team delivered the rescued people to a certain safe spot, it will be dispatched to pick up the next person that needs rescue directly or drive back to the dispatching center to stand by.

A. Framework of MobiRescue

MobiRescue runs periodically (e.g., every 5 minutes) to dispatch the rescue teams based on the predicted distribution of potential rescue requests. *MobiRescue* consists of the following stages as shown in the three dashed boxes in Figure 7:

1. Human mobility information derivation. First, we apply the *Data Cleaning* (e.g., filtering out positions out of the actual range of our interested city, redundant positions). Then, based on the cleaned data, we derive the *Trajectories in Landmarks* of humans on the *Roadmap with Landmarks and Road Segments* as explained in Section III-A.

2. Predicting distribution of potential rescue requests (Section IV-B). Based on the output of *Trajectories in Landmarks* from the first stage, we input the *Disaster-related Factors* (i.e., precipitation, wind speed, altitude) of each road segment to the *SVM Model* to predict the *Distribution of Potential Rescue Requests* across all the road segments.

3. Reinforcement Learning based Rescue Team Dispatching (Section IV-C). Based on the *Distribution of Potential Rescue Requests* and the remaining available road network (denoted as $\widetilde{G} = (\widetilde{E}, \widetilde{V})$) obtained from the satellite imaging from National Weather Service [11], we train and utilize the *RL based Rescue Team Dispatching* method to decide the road segment each rescue team should drive to in order to maximize the total number of fulfilled rescue requests, minimize the rescue teams' driving delay to the rescue requests' positions, and minimize the number of serving rescue teams.

B. Predicting Distribution of Potential Rescue Requests

From Observation 1 (Section III-B1), we know that although the overall city area was greatly affected by a disaster, the disaster has different impact severities on different regions due to their different disaster-related factors. Based on the impact severity and surrounding disaster-related factors, people send out rescue requests. For example, under a flooding disaster, for people that are trapped in low altitude places, they must be rescued as soon as possible because their basic living is already or will be in danger. In addition to people's altitude, more disaster-related factors (e.g., precipitation, wind speed, altitude) need to be considered in general disaster situations. Thus, one research problem is: *Given the distribution of people in an area affected by disaster, how to determine the distribution of potential rescue requests with the consideration of various disaster-related factors?* In this section, we present the details of our SVM [10] based model to solve this problem.

According to previous works on the analysis of disaster characteristics [21], [22], the primary disaster-related factors that affect the living conditions of people in post-disaster situations can be categorized into: (*precipitation*, *wind speed*, *altitude*) for hurricane, flooding or tsunami, and (*seismic magnitude*, *altitude*, *building density*) for earthquake. In this paper, we focus on hurricane related factors represented as a vector $\mathbf{h} =$ (*precipitation*, *wind speed*, *altitude*). *Precipitation* and *wind speed* can be obtained from the external facilities such as National Weather Service. *Altitude* can be obtained from the altimeter sensor on the person's cellphone [23]. Thus, each person will have a vector of the factors corresponding to his/her position. For example, if a person is on a playground, the vector of the person's surrounding disaster-related factors is (100 mm, 10 mph, 5 m). If the person is moved to a safe spot, the vector of the person's surrounding disaster-related factors changes to (10 mm, 5 mph, 50 m).

To classify whether a person should be rescued based on the disaster related factors surrounding this person, an effective approach is to use machine learning technique to take the person's disaster related factors as inputs and output the person's rescue decision (i.e., should be rescued or should not be rescued). By combining all the persons that should be rescued, we can predict the distribution of potential rescue requests. Specifically, we choose the SVM model for the classification. The SVM is a classifier model that maps the feature data (i.e., disaster-related factors in this paper) as points in a high-dimensional space. After training, SVM determines a *hyperplane* that separates the data points into the two types of rescue decisions and the corresponding weights of the factors, so that the distance from the *hyperplane* to the nearest data points on each side is maximized. The reasons of selecting SVM as the classifier are as follows:

- The disaster-related factors (precipitation, wind speed, altitude) are features with continuous numerical values, which are suitable for the processing of SVM.
- We aim to do binary classification on the people's rescue decision (i.e., should be rescued or should not be rescued), which can be achieved by the SVM classifier.
- In a high-dimensional space, the data points may not be separated with a linear function. The SVM classifier can overcome this problem by using the kernel function, which supports the formulation of a nonlinear function for separating the data points in training the classification model.

To get the training data, for each person in the historical movement data during disaster, we first checked whether the person was rescued by using the method of identifying rescued people introduced in Section III-B2. We consider this as the ground truth of the person's rescue requests. Then, we located the person's previous staying position before he/she was delivered to a hospital and calculated the disaster-related factor vector of the person.

Finally, we input the disaster-related factor vector of each person in the dataset to the SVM model and train the model to output the corresponding rescue decision for the person with the minimum error. Once the training of the SVM model is complete, given the real-time disaster-related factor vector of a person as the inputs, we can utilize the SVM model to determine whether the person should be rescued. Specifically, we use Equation (1) to represent the decision of the SVM model on a person q's rescue decision:

$$f(p_q, \mathbf{h}_q) = \begin{cases} 1, & \text{person } q \text{ should be rescued,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where f is the trained SVM model, p_q is the position of person q, and $\mathbf{h}_q$ is the disaster-related factor vector at q's position. After applying the SVM model to determine the rescue decision of each person, we count the number of persons that need to be rescued on each road segment $e_i \in E$ (i.e., distribution of potential rescue requests on road segments) as

$$\widetilde{n}_{e_i} = \sum_{q \in Q_{e_i}} f(p_q, \mathbf{h}_q), \tag{2}$$

where Q_{e_i} is the set of people on road segment e_i. $\widetilde{n}_{e_i}$ will be used in the reinforcement learning based rescue team dispatching method introduced below.

C. Reinforcement Learning based Rescue Team Dispatching

From Observation 2 (Section III-B2), we know that the distribution of people's movement has significant change before, during and after the disaster. The previous integer programming based rescue team dispatching methods are time-consuming and cannot efficiently guide the rescue teams in real time to serve the people's rescue requests. Thus, we need a real-time rescue team dispatching method to maximize the number of fulfilled rescue requests, and meanwhile minimize the rescue teams' driving delay to the rescue requests' position and the total number of serving rescue teams.

Therefore, we propose the RL based dispatching method to determine the action of all the rescue teams. RL can output the guidance for the rescue teams much faster than the integer programming based methods. Therefore, the guidance of the rescue teams can be produced in real time. This method runs based on the predicted distribution of potential rescue requests from Section IV-B periodically (e.g., 5 minutes). The dispatching process of the RL model works as follows: given a current *state* s, the model outputs an *action* a that maximizes the *reward* r. As shown in Figure 8, as state, *Current Distribution of Potential Rescue Requests*, the *Rescue Teams' Current Positions* are the inputs to the RL model, which outputs the *Action* of all the rescue teams. The *action* of a rescue team includes driving to a road segment to pick up a person that needs rescue or driving back to the dispatching center to stand by. The *reward* resulted by an action is defined as the weighted sum of the total number of rescue requests

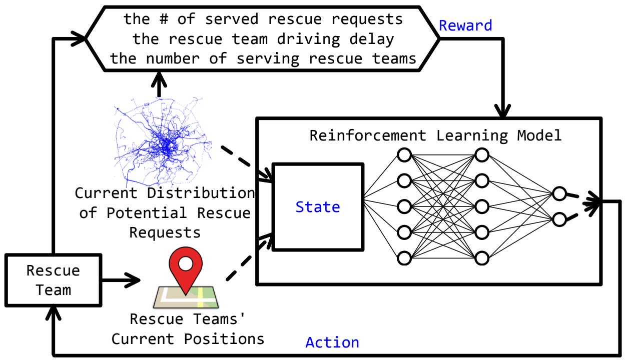

the # of served rescue requests
the rescue team driving delay
the number of serving rescue teams
Reward
Reinforcement Learning Model
State
Current Distribution of Potential Rescue Requests
Rescue Team
Rescue Teams' Current Positions
Action

Fig. 8: Training of reinforcement learning model.

that the dispatched rescue teams fulfilled, the total sum of the dispatched rescue teams' driving delays, and the number of serving rescue teams. Once we optimize the RL model's policy $\pi : s \mapsto a$ through training, we can use it to optimally guide the rescue teams' movement that maximizes the reward. The guided rescue teams will serve the people's rescue requests appearing on their driving routes. Next, we introduce the details of the state, action and reward of the RL model, as shown in Figure 8.

1) State: One goal is to maximize the total number of rescue requests fulfilled by all the rescue teams, and each rescue team can only select one road segment from the road network as its destination at a time. Therefore, the state needs to include *current distribution of potential rescue requests*, i.e., the estimated number of potential rescue requests on each road segment $\widetilde{n}_{e_i}$ calculated by Equation (1). Specifically, recall that for each road segment in the remaining available road network $e_i \in \widetilde{E}$, we utilize the SVM model (Equation (1)) to determine the number of potential rescue requests on e_i (denoted by $\widetilde{n}_{e_i}$). Another goal is to minimize the rescue teams' driving delay from their current positions to the rescue requests' positions, so the state also needs to include the *rescue teams' current positions*. In summary, the state s consists of current position of each rescue team and the estimated number of potential rescue requests on each road segment in the remaining available road network. Thus, the state set S is defined as follows:

$$s = (\{p_{m_k} \mid \forall\, m_k \in \mathcal{M}\}, \{\widetilde{n}_{e_i} \mid \forall\, e_i \in \widetilde{E}\}) \in S \tag{3}$$

where m_k denotes the k^{th} rescue team, p_{m_k} denotes the current position of the k^{th} rescue team, and $\mathcal{M}$ denotes the set of all the rescue teams.

2) Action: The action consists of the driving decision of each rescue team. Specifically, the k^{th} rescue team's driving decision (denoted by x_{m_k}) is: i) driving to a destination road segment in the remaining available road network (denoted by $x_{m_k}{=}e_j \in \widetilde{E}$), or ii) driving to the rescue team dispatching center to stand by (denoted by $x_{m_k} = 0$). The rescue teams that are chosen to drive to a certain destination road segment are considered as serving rescue teams. Thus, the action set A is defined as:

$$a = (x_{m_k} \mid \forall\, m_k \in \mathcal{M}) \in A. \tag{4}$$

3) Reward: As aforementioned, the RL model aims to maximize the total number of rescued people, and meanwhile minimize the sum of all the rescue teams' driving delays and the number of serving rescue teams. The reward function resulted by the x_{m_k} of all the rescue teams is defined as the weighted sum of the total number of rescue requests actually served by all the rescue teams (denoted by N^q), the sum of the rescue teams' driving delays to the rescue requests' positions (denoted by T^d), and the number of serving rescue teams (denoted by N^m), which can be formulated as:

$$r(s_t, a_t, s_{t+1}) = \alpha N^q - \beta T^d - \gamma N^m \tag{5}$$

where s_t is the state at current time, s_{t+1} is the next state caused by the action a_t, and α, β and γ are the weights that determine the importance of the metrics that can be manually set.

The total number of served rescue requests metric is the total number of rescue requests that all the rescue teams may encounter by driving to their respective destination road segments. Specifically, for each rescue team (say k^{th} rescue team), we first use an existing routing algorithm (e.g., the Dijkstra algorithm [19]) to determine the shortest distance driving route from the rescue team's current position (p_{m_k} in the state) to the end of the destination road segment in the determined action ($e_j \in \widetilde{E}$), which is denoted as $\Phi_{kj} = \{p_{m_k}, \ldots, e_j\}$. Then we calculate the union set of the road segments that will be driven by all the rescue teams by taking their actions (denoted by $E' = \bigcup_{m_k \in \mathcal{M}} \Phi_{kj}$). Finally, we sum the numbers of the served rescue requests in each road segment in E' (denoted by n_{e_i}): $N^q = \sum_{e_i \in E'} n_{e_i}$.

The rescue team driving delay metric is the sum of all the serving rescue teams' driving delays $T^d = \sum_{m_k \in \mathcal{M}} t_{kj}$, where t_{kj} is the driving delay of the k^{th} rescue team. Specifically, if the k^{th} rescue team is chosen to drive to the end of road segment e_j (denoted by $x_{m_k} = e_j \in \widetilde{E}$), its driving delay to e_j is calculated by $t_{kj} = \sum_{e_i \in \Phi_{kj}} \frac{l_{e_i}}{v_{e_i}}$, where l_{e_i} is the length of e_i, and v_{e_i} is the speed limit of e_i under current flooding disaster condition.

The number of serving rescue teams metric is the number of rescue teams that are chosen to drive to certain destination road segments. Specifically, if a rescue team m_k is chosen to drive to a new destination road segment to serve the rescue requests ($x_{m_k} = e_j \in \widetilde{E}$), it is counted as a serving rescue team. That is, $c_{m_k} = 1$. If rescue team m_k is chosen to drive to the dispatching center ($x_{m_k} = 0$), it is not counted as a serving rescue team. That is, $c_{m_k} = 0$. Thus, this metric is calculated as $N^m = \sum_{m_k \in \mathcal{M}} c_{m_k}$.

4) Obtaining the Optimal Policy: We utilize the Deep Neural Network (DNN) (as in [24]) to obtain the optimal policy for dispatching rescue teams. For DNN training, we get the historical distribution of rescue requests and the historical positions of rescue teams from previous disasters with a short sampling interval (e.g., 1 minute) as the training data. After the completion of model training, we can utilize the output of the model to generate a routing plan which can guide each rescue team during disaster. However, this sampled historical data may not be completely applicable for the current disaster area since historical disasters may have different levels of impact.

Therefore, once the RL model is trained and when running the RL model, we keep training the RL mode. That is, we keep getting the up-to-date training data with a short sampling interval by recording the generated rescue teams' driving routes as the output data (i.e., action of all the rescue teams), and the position of each rescue team and the distribution of rescue requests on all the road segments as the input data (i.e., state of all the rescue teams).

5) Extension for General Catastrophic Situations: MobiRescue can be extended to handle different catastrophic situations. Specifically, the following components can be more sophisticatedly designed for the extension. We leave the detailed extension of the components as our future work.

1. **Disaster-related factors.** It is worth clarifying that the disaster-related factors are not necessarily limited to the ones considered in this paper (i.e., precipitation, wind speed, altitude), but should be selected according to different types of disasters. For example, (temperature, precipitation, wind direction) may be more effective factors for victim prediction related to wild fire hazard; (precipitation, road slope degree, temperature) may be more effective factors for damage assessment under icy or snowy weather. We leave the adaptive selection of the disaster-related factors according to different types of disaster as our future work directions.
2. **Availability of real-time GPS data.** Under severe situations, the GPS locations of some people may not be readily available. We can refer to these people's historical GPS data to analyze the home address/work address/preferred driving pattern and estimate the approximate position/area of the people.

V. Performance Evaluation

A. Comparison Methods

We evaluate the performance of *MobiRescue* (*MR* in short) in comparison with a representative emergency vehicle dispatching method for normal situations [5] (*Schedule* in short) and a representative rescue team dispatching method for catastrophic situations [8] (*Rescue* in short). *Schedule* uses integer programming to dispatch emergency vehicles on demand of the appearance of emergency requests to minimize the emergency vehicles' average driving delay to the emergency event positions. However, it aims to serve appearing emergency requests under normal situations but is not applicable for the flooding disasters since they cannot predict the appearance of rescue requests and proactively guide the rescue teams to timely serve the requests. Long time in solving the integer programming problem further prevents the timely rescue serving.

Rescue applies time series analysis on historical distribution of rescue request appearances. It predicts the rescue request demand at the current hour by using the weighted average request demand at this hour in several previous days. Based on the predicted distribution of rescue requests, it formulates an integer programming problem to minimize the sum of the rescue teams' driving delays to the predicted rescue requests' positions and periodically solves the problem to update the rescue teams' driving route according to the changed distribution of potential rescue requests. The drawback of this method is that it does not consider the factors (e.g., precipitation, wind speed, altitude) that reflect the danger level of people's surrounding environment, which causes insufficient accuracy in estimating the positions of potential rescue requests. Also, solving the integer programming problem is time-consuming and cannot efficiently guide the rescue teams in real time.

For fair comparison, we suppose the deployment of hospitals in the three methods follows the deployment of existing hospitals in Charlotte city, and all three systems deliver the rescued people to their nearest hospitals.

B. Experiment Settings

We use the human mobility data in Charlotte on September 16, 2018 to simulate the appearance of rescue requests. This day is representative because that it has the highest number of rescue requests and the requests are distributed in different regions according to the measurement made in Section III-B1. We use the data from the Hurricane Michael (October 7-16, 2018), which also impacted the Charlotte area, to train our SVM model and RL model. For the SVM model's training, we got the rescue decision on each person using the method introduced in Section III-B2 as the ground truth. The disaster-related factors are determined based on the weather data in Charlotte during the same time period. For the RL model's training, we suppose that the initial positions of rescue teams (i.e., ambulances) were randomly distributed among all the hospitals. The number of ambulances is equal to the maximum daily number of requests over all days during the hurricane. Given the hospital delivery timestamp (t_d) of a person in our ground truth, we chose the ambulance that can drive to the person's request position and deliver him/her to the destination hospital with the estimated delivery timestamp most approximate to t_d as the ambulance that picked up the person. We also suppose that after each rescue, the ambulance chose to stay in its nearest hospital. Then, we use the driving routes of all the ambulances as the output. Meanwhile, we sample the position of each rescue team per unit time (e.g., 1 minute) and the ground truth historical distribution of rescue requests on all the road segments per unit time as the inputs.

We utilize Flow [25], which is a computational framework for RL and control experiments for traffic simulation, to train the RL based rescue team dispatching method. Based on the deployment of existing hospitals in Charlotte, we use SUMO [26], which is a simulator for city-scale traffic simulation, to simulate the movement of 100 rescue teams for 24 hours on Charlotte's road network. We call the rescue requests that are served within 30 minutes *timely served requests*. We converted OpenStreetMap road network of Charlotte to a SUMO road network file. The metrics we measured include:

- *The total number of timely served rescue requests*: For each rescue team, we measure the number of rescue requests it has timely served in each hour throughout a day. Then, we measure the total number of timely served rescue requests of all the rescue teams in each hour throughout the day. We also measure the CDF of the numbers of timely served rescue requests of all the rescue teams throughout the day. The purpose of this metric is to compare the performance of different methods in serving the rescue requests.
- *The average driving delay*: For each rescue request from a person, we measure the rescue team's driving delay to his/her

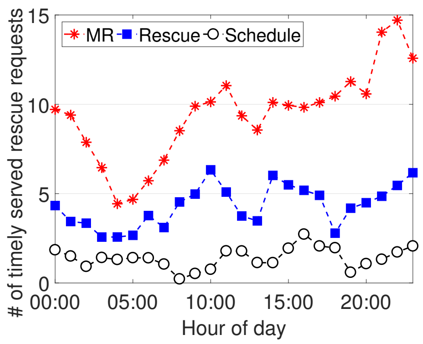


Fig. 9: The total number of served rescue requests.

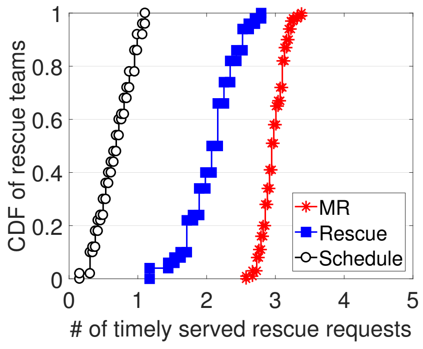


Fig. 10: CDF of the numbers of served rescue requests of rescue teams.

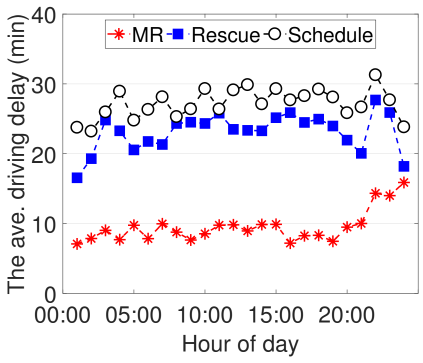


Fig. 11: The average driving delay.

position. Then, we measure the average driving delay over all the served rescue requests in each hour throughout a day. We also measure the CDF of the driving delays of all the served rescue requests throughout the day. The purpose of this metric is to compare the performance of different methods in reducing the waiting time of the rescue requests before being served by a rescue team.

• *Timeliness of rescuing*. For each rescue request from a person, we measure (the person's rescue time - person's request time) as the timeliness of rescuing. For the case when rescue team has already arrived at the person's position before the actual request, we set the timeliness of rescuing as 0. The purpose of this metric is to compare the timeliness performance of different methods in dispatching the rescue teams. Note that the computation delay resulted from the dispatching methods is included in this metric.

• *The number of serving rescue teams*. During each hour throughout a day, we measure the number of serving rescue teams. The purpose of this metric is to compare the performance of different methods in reducing the cost of dispatching.

• *The prediction accuracy and precision of rescue requests*: To demonstrate the performance of our SVM based method in predicting whether a person needs rescue, for each road segment, we measure the prediction accuracy as $\frac{TP+TN}{TP+TN+FP+FN}$, and measure the prediction precision as $\frac{TP}{TP+FP}$, where *True Positive (TP)* = the number of people correctly predicted as sending rescue requests; *False Positive (FP)* = the number of people incorrectly predicted as sending rescue requests; *True Negative (TN)* = the number of people correctly predicted as not sending rescue requests; *False Negative (FN)* = the number of people incorrectly predicted as not sending rescue requests.

C. Experimental Results

1) The Total Number of Timely Served Rescue Requests: Figure 9 shows the total number of timely served rescue requests of all the rescue teams during each hour throughout a day under different methods. Figure 10 shows the CDF of the numbers of timely served rescue requests of all the rescue teams throughout the day under different methods. We can see that the results of *MobiRescue* are always the highest compared with the other methods. The results follow: *MobiRescue*>*Rescue*>*Schedule*.

In *Schedule*, the rescue teams' driving routes are updated on demand of the appearance of real-time rescue requests. Based on the real-time distribution of rescue requests, *Schedule* uses integer programming to minimize the rescue teams' average driving delay to the appearance positions of rescue requests. However, *Schedule* aims to serve appearing emergency requests under normal situations but is not applicable for the flooding disaster situations since they cannot predict the appearance of rescue requests and proactively guide the rescue teams to timely serve the requests. Long time in solving integer programming problem further prevents its timely rescuing.

Rescue applies time series analysis on historical distribution of rescue request appearances to periodically predict the future appearance of people's rescue requests and periodically solving its integer programming to update the driving routes of the rescue teams. However, it does not consider the disaster-related factors that closely reflect the danger level of people's surrounding environment, which causes insufficient accuracy in estimating the positions of potential rescue requests. Also, when planning the driving route of a rescue team, *Rescue* needs to spend a long time on solving its integer programming problem. Therefore, *Rescue* cannot efficiently guide the rescue teams in real time to serve the rescue requests, which causes much fewer served rescue requests than that of *MobiRescue*.

In *MobiRescue*, the rescue team dispatching center monitors the disaster-related factors in real time and utilizes the SVM based method to periodically estimate the distribution of potential rescue requests with a high accuracy. Based on the predicted distribution of potential rescue requests, *MobiRescue* can avoid the rescue teams from driving to regions that have low likelihood of rescue request appearance. Also, the periodical running of the RL model can always guide the rescue teams in real time to cruise around the areas with the highest possibility of served potential rescue requests and reduce rescue

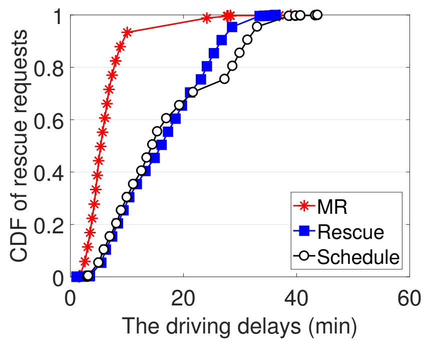


Fig. 12: CDF of the driving delays.

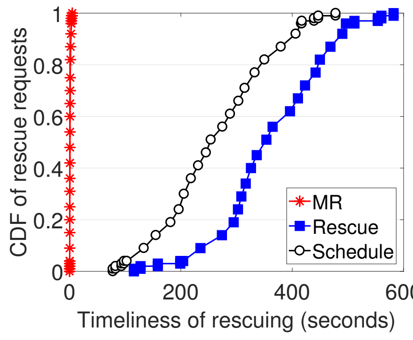


Fig. 13: Timeliness of rescuing.

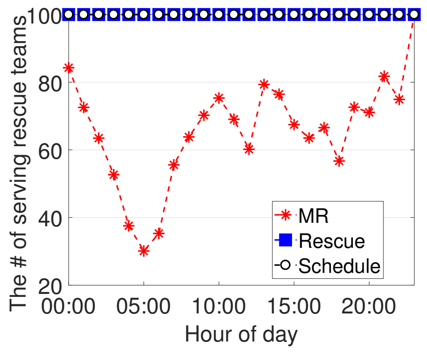


Fig. 14: Num. of serving rescue teams.

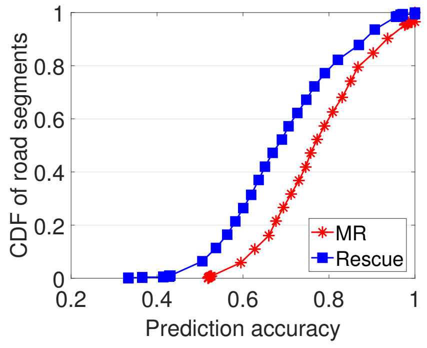


Fig. 15: CDF of prediction accuracies of rescue requests on road segments.

teams' driving time. Therefore, the rescue teams of *MobiRescue* can serve the most rescue requests among the methods.

2) The Average Driving Delay to the Rescue Requests' Positions: Figure 11 shows the average driving delays of the rescue teams to serve the rescue requests during each hour throughout a day under different methods. Figure 12 shows the CDF of the driving delays of all the served rescue requests throughout the day under different methods. We can see that the results of *MobiRescue* is much lower than the other two methods during most times. The results follow: *MobiRescue*<*Rescue*<*Schedule*. *Schedule* does not consider the real-time road network connection status under flooding disaster condition, which causes the emergency vehicles to waste time on routes with unavailable road segments. Therefore, it results in the longest average driving delay. In *Rescue*, the estimated positions of potential rescue requests output by its time series analysis are not sufficiently accurate since it does not consider the factors that reflect the danger level of people's surrounding environment. The rescue teams wasted some time on driving to some inaccurate request positions. Therefore, it results in the second longest average driving delay. In contrast, in *MobiRescue*, the rescue team dispatching center can utilize the SVM based method to periodically estimate the distribution of potential rescue requests with a high accuracy, and aims to minimize the total driving delays of all rescue teams based on the real-time connection status of remaining available road network. The rescue teams can approach the accurate request positions through available road segments. Therefore, it results in the shortest average driving delay.

3) Timeliness of Rescuing: Figure 13 shows the CDF of the timeliness of rescuing of all the rescue requests under different methods. We can see that the results follow: *MobiRescue*<<*Schedule*<*Rescue* in all dispatchings. This is because that in *Rescue* and *Schedule*, solving the integer programming problem generally takes around 300 seconds for computation and the computation time varies under different amounts of request demands (i.e., the more requests, the more complex for solving the integer programming problem). Therefore, the long computation time in solving their integer programming problem prevents their timely rescue serving. While in *MobiRescue*, once the RL model training is complete, it takes less than 0.5 second in computing and outputting the dispatching guidance. This result confirms that *MobiRescue* can achieve real-time dispatching of rescue teams compared to integer programming based dispatching methods.

4) The Number of Serving Rescue Teams: Figure 14 shows the number of serving rescue teams during each time slot throughout a day under different methods. We can see that the results follow: *MobiRescue*<*Rescue*=*Schedule*. In *Rescue* and *Schedule*, the number of serving rescue teams remains constant during all time slots since their integer programming based methods do not aim to minimize the number of serving rescue teams according to the change of rescue request appearance. In *MobiRescue*, the change of the number of serving rescue teams generally matches the change of the number of rescue request appearances as illustrated in Figure 9. This is because that the RL based rescue team dispatching method aims to minimize the total number of serving rescue teams in covering the rescue requests. This result confirms that *MobiRescue* can reduce the cost of dispatching rescue teams of the other methods.

5) The Prediction Accuracy and Precision of Rescue Requests: Figure 15 shows the CDF of the prediction accuracies of rescue requests on all the road segments of *MobiRescue* and *Rescue*. *Schedule* is not included since it does not have such

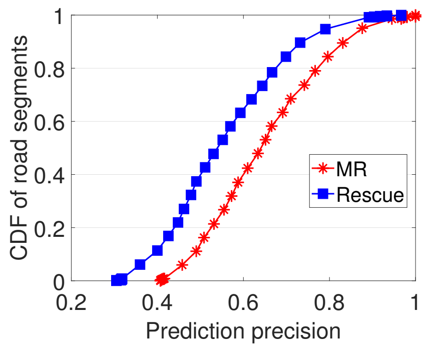


Fig. 16: CDF of prediction precisions of rescue requests on road segments.

prediction function. Figure 16 shows the CDF of the prediction precisions of rescue requests on all the road segments. We can see that the results follow: *MobiRescue*>*Rescue* for all road segments. This is primarily because that the time series analysis method in *Rescue* does not consider the factors that reflect the danger level of people's surrounding environment. In *MobiRescue*, since the SVM model learns the impacts of the factors during the training on historical rescue request records, *MobiRescue* can achieve both higher prediction accuracy and higher prediction precision over all the road segments since it additionally considers the disaster-related factors.

VI. Conclusion

Rescue team dispatching under a flooding disaster is crucial for decreasing the number of deaths and injuries. Previous emergency vehicle dispatching methods in normal situations cannot effectively handle flooding disaster situations, and previous rescue team dispatching methods have insufficient accuracy in predicting the distribution of rescue requests by only relying on the historical data that may not accurately reflect current disaster impact in different regions and cannot guide the rescue teams in real time since their integer programming based methods are time-consuming. Our proposed *MobiRescue* is the first human mobility based rescue team dispatching method that utilizes SVM model and Reinforcement Learning to maximize the total number of served rescue requests, minimize the rescue teams' driving delay to the rescue requests' positions and the number of serving rescue teams. Our analytical results on a city-scale human mobility dataset provide foundation for the design of *MobiRescue*. We develop an SVM based method that utilizes the disaster-related factors of regions to predict the distribution of potential rescue requests. Based on the predicted distribution of potential rescue requests, we develop a Reinforcement Learning based rescue team dispatching method to achieve the aforementioned goals. We conducted trace-driven experiments on SUMO and Flow to verify the superior performance of *MobiRescue* over other representative comparison methods. In the future, we plan to focus on building a rescue request prediction model that considers more disaster-related factors under various catastrophic situations.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1827674, CCF-1822965, OAC-1724845, and Microsoft Research Faculty Fellowship 8300751.

References

[1] F. Cavdur, M. Kose-Kucuk, and A. Sebatli, "Allocation of temporary disaster response facilities under demand uncertainty: An earthquake case study," *IJDRR*, vol. 19, 2016.
[2] A. Edrissi, M. Nourinejad, and M. J. Roorda, "Transportation network reliability in emergency response," *TRPE: logistics and transportation review*, vol. 80, 2015.
[3] V. Schmid, "Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming," *European journal of operational research*, vol. 219, no. 3, 2012.
[4] M. S. Maxwell, M. Restrepo, S. G. Henderson, and H. Topaloglu, "Approximate dynamic programming for ambulance redeployment," *INFORMS Journal on Computing*, vol. 22, no. 2, 2010.
[5] P. L. Van Den Berg, J. T. Van Essen, and E. J. Harderwijk, "Comparison of static ambulance location models," in *Proc. of GOL*, 2016.
[6] L. V. Snyder and M. S. Daskin, "Reliability models for facility location: The expected failure cost case," *Transportation Science*, vol. 39, no. 3, 2005.
[7] B. Sun, W. Ma, and H. Zhao, "A fuzzy rough set approach to emergency material demand prediction over two universes," *Applied Mathematical Modelling*, vol. 37, no. 10-11, 2013.
[8] M. Huang, K. R. Smilowitz, and B. Balcik, "A continuous approximation approach for assessment routing in disaster relief," *TRPB: Methodological*, vol. 50, 2013.
[9] S. El-Tawab, M. Abuelela, and Y. Gongjun, "Real-time weather notification system using intelligent vehicles and smart sensors," in *Proc. of MASS*, 2009.
[10] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, 1995.
[11] "National weather service," https://www.weather.gov/phi/localclimate, accessed in April, 2019.
[12] L. Yan, S. Mahmud, H. Shen, N. Z. Foutz, D. E. Brown, W. Yusuf, D. Loftis, L. Lyons, J. L. Goodall, and J. Anton, "MobiAmbulance: Optimal scheduling of emergency vehicles in catastrophic situations," in *Proc. of ICCCN*, 2019.
[13] "Openstreetmap," http://www.openstreetmap.org/, accessed in April, 2019.
[14] "Apache Hadoop," http://hadoop.apache.org/, accessed in April, 2019.
[15] "Apache spark," http://spark.apache.org/, accessed in April, 2019.
[16] L. Yan, H. Shen, Z. Li, A. Sarker, J. A. Stankovic, C. Qiu, J. Zhao, and C. Xu, "Employing opportunistic charging for electric taxicabs to reduce idle time," *ACM IMWUT*, vol. 2, no. 1, 2018.
[17] L. Yan, H. Shen, J. Zhao, C. Xu, F. Luo, and C. Qiu, "CatCharger: Deploying wireless charging lanes in a metropolitan road network through categorization and clustering of vehicle traffic," in *Proc. of INFOCOM*, 2017.
[18] G. Wang, X. Xie, F. Zhang, Y. Liu, and D. Zhang, "bCharge: Data-driven real-time charging scheduling for large-scale electric bus fleets," in *RTSS*, 2018.
[19] Y. Zheng, "Trajectory data mining: an overview," *TIST*, 2015.
[20] K. Pearson, "Note on regression and inheritance in the case of two parents," in *Proc. of the Royal Society of London*, 1895.
[21] L. Chen and E. Miller-Hooks, "Optimal team deployment in urban search and rescue," *TRPB: Methodological*, vol. 46, no. 8, 2012.
[22] L. Zhou, X. Wu, Z. Xu, and H. Fujita, "Emergency decision making for natural disasters: An overview," *IJDRR*, vol. 27, 2018.
[23] C. Qiu and A. C. Squicciarini, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," in *Proc. of ICDCS*, 2019.
[24] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. of SIGCOMM*, 2017.
[25] C. Wu, A. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen, "Flow: Architecture and benchmarking for reinforcement learning in traffic control," *arXiv*, 2017.
[26] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *IJASM*, vol. 5, no. 3&4, 2012.