The Thirty-Second Innovative Applications of Artificial Intelligence Conference (IAAI-20)

Improving Efficiency of Volunteer-Based Food Rescue Operations

Zheyuan Ryan Shi,[1*] **Yiwen Yuan,**[1*] **Kimberly Lo,**[1] **Leah Lizarondo,**[2] **Fei Fang**[1]
[1]Carnegie Mellon University, [2]412 Food Rescue
ryanshi@cmu.edu, {yiweny, kimberll}@andrew.cmu.edu, leah@412foodrescue.org, feif@cs.cmu.edu

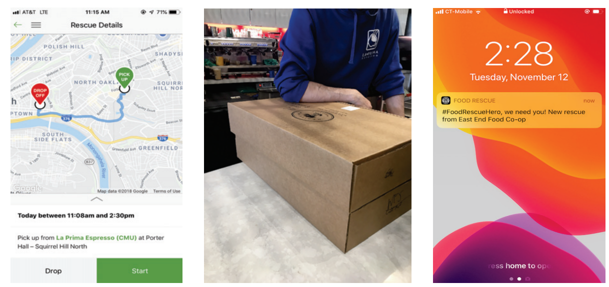
Abstract

Food waste and food insecurity are two challenges that coexist in many communities. To mitigate the problem, food rescue platforms match excess food with the communities in need, and leverage external volunteers to transport the food. However, the external volunteers bring significant uncertainty to the food rescue operation. We work with a large food rescue organization to predict the uncertainty and furthermore to find ways to reduce the human dispatcher's workload and the redundant notifications sent to volunteers. We make two main contributions. (1) We train a stacking model which predicts whether a rescue will be claimed with high precision and AUC. This model can help the dispatcher better plan for backup options and alleviate their uncertainty. (2) We develop a data-driven optimization algorithm to compute the optimal intervention and notification scheme. The algorithm uses a novel counterfactual data generation approach and the branch and bound framework. Our result reduces the number of notifications and interventions required in the food rescue operation. We are working with the organization to deploy our results in the near future.

1 Introduction

In the US, over 25% of the food is wasted, with an average American wasting about one pound of food per day (Conrad et al. 2018). Meanwhile, 11.8% of American households struggle to secure enough food at some point (Coleman-Jensen et al. 2018). Among the several responses to this inefficient food distribution, food rescue organizations are emerging in many cities. They receive edible food from restaurants and groceries ("donors") and send it to organizations serving low-resource communities ("recipients"). These food rescue organizations are an important force to fight against food waste and food insecurity, both included in the United Nations' Sustainable Development Goals.

A food rescue organization functions as a platform between the donors and the recipients. Upon receiving the notice from a donor, the organization matches the food to a recipient. Typically, it transports the food from the donor to the recipient, or stores the food at its own facility if necessary. This incurs cost and there are existing works on optimizing the matching process to minimize this cost (Nair et al. 2018), and some attempt to create a market (Prendergast 2016). However, many of these organizations operate under tight budget and human resource constraints. As a result, some outsource the transportation of food to local volunteers, which brings in a new dimension to the problem.



(a) Screenshot of a rescue task (b) Donation picked up at a cafe (c) App notification

Figure 1: 412 Food Rescue operations

We collaborate with 412 Food Rescue (412FR), a food rescue organization serving over 1000 donors and recipient organizations in Pittsburgh, US. The dispatcher at 412FR matches the food by calling each recipient till some recipient accepts. The dispatcher determines the order of these calls based on numerous factors such as the proximity between the donor and recipient and the estimated recipient's willingness to accept the food. This decision is not hard-coded but depends on the dispatcher's rich experience. After the matching, they post the rescue on 412FR's smartphone apps. 412FR's over 7000 volunteers can see the rescue's start and end location as well as the weight and type of food (Fig. 1a). A volunteer can claim the rescue on the app and then complete the rescue by picking up the food from the donor within its pickup window (Fig. 1b) and delivering it to the recipient.

Relying on volunteers saves cost for the organization, but there is a high degree of uncertainty in whether a rescue will be claimed and completed. Over the years, 412FR has used many methods to get more rescues claimed and completed.

[*]Z. R. Shi and Y. Yuan contributed equally to this work.

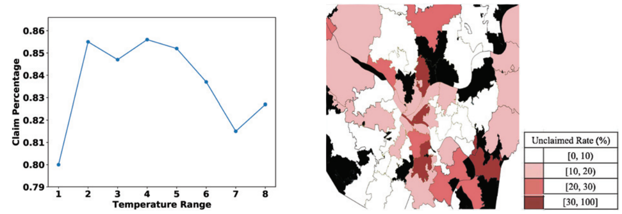
First, after a rescue request is posted, the app will push notification to volunteers within 5 miles (Fig. 1c). After 15 minutes, if no one has claimed the rescue, the app will push notification to all available volunteers. Second, dispatcher monitors all rescues to be claimed or to be completed. If no one has claimed a rescue by the last hour of its pickup window, the dispatcher calls *regular* volunteers they are familiar with to help with the rescue. For the ones claimed and to be completed, dispatcher needs to answer volunteers' inquiries about delivery details in real-time. As such, the dispatcher has a heavy workload. However, while it is helpful to engage with the volunteers, too many notifications might drive them away (Felt, Egelman, and Wagner 2012).

In this paper, we aim to reduce the dispatcher's workload and the redundant notifications sent to the volunteers, without decreasing the claim rate of the rescues. We make two main contributions. 1) We train a stacking model to predict whether a rescue will be claimed. Our stacking model achieves an AUC of 0.81, serving as a reliable reference of the risk of a rescue. The model informs the dispatcher how likely a rescue is going to be claimed, thus helping the dispatcher better plan for backup options. (2) We perform data-driven optimization to find the optimal *Intervention and Notification Scheme* (INS), i.e., when the dispatcher should intervene and seek help from regular volunteers and when and to whom the notifications should be sent. We estimate the counterfactual rescue outcomes and use a branch and bound method to improve computational efficiency. The resulting INS can improve over the current practice by reducing the number of notifications sent and the dispatcher interventions, while keeping the rescues' expected claim rates. Our analysis suggests to the platform some changes in their current INS, which can save the most valuable resources to food rescue: the dispatcher's attention and volunteers' interest.

We are working with 412FR to deploy our results. In fact, such organizations are not rare at all. In the US alone, similar organizations are already operating in over 55 cities, helping over 11 million people, and the numbers will only keep growing. Thus, our work could potentially improve the dispatching decisions at a large scale, not to mention the similar volunteer-based community services other than food rescue.

2 Related Work

The operational challenges of food rescue organizations have received much attention. Nair et al. (2018) and Gunes, van Hoeve, and Tayur (2010) study matching the donor and recipient with a routing problem. This is related to the more general problem of online matching (Karp, Vazirani, and Vazirani 1990; Mehta, Waggoner, and Zadimoghaddam 2015). Prendergast (2016) and Lundy et al. (2019) consider the incentive of the agencies and design a market for the food rescue platform. Phillips, Hoenigman, and Higbee (2011) explore predicting the future donations. However, all these works assume the organization manages the donations without the participation of volunteers, and thus they are not applicable to our problem. In addition, the well-studied task allocation problem (Ho and Vaughan 2012) does not perfectly fit our scenario, as 412FR has no control over the volunteers. The only work which assumes similar operation is (Lee et al.



(a) Claim rate vs. temperature. (b) Percentage of unclaimed rescues by zip code district.

Figure 2: Data analysis results. The temperature range i represents $(10.5i - 11.5, 10.5i - 1]$°F.

2019). It studies the stakeholders' perception of fairness and democracy on the food matching decisions, while we focus on improving the efficiency of food rescue operations.

Our prescriptive analysis uses counterfactual estimation of rescue outcomes under various dispatching schemes. This relates to the extensive literature on causal inference with observational data (Dehejia and Wahba 2002). However, 412FR has always used the same dispatching scheme for all rescues, and it is currently impossible to contact volunteers for pre- and post-intervention tests (Pratt, McGuigan, and Katzev 2000). Thus, existing work is not applicable and we develop a new way of constructing counterfactual datasets.

3 Predicting the Claim of Rescues

Our first task is to predict whether a rescue would be claimed. We use the operational dataset of 412FR which contains rescues from March 2018 to May 2019. The dataset records the time log of each step in the rescue: posting, claimed by volunteer, and completion, along with the ID of the volunteer who claimed the rescue. We treat a rescue as unclaimed and assign a negative label if it was never claimed or if it was claimed within the last hour of the pickup window by a selected group of volunteers who had done more than 10 rescues within the last two months. We assume the latter ones had gone through dispatcher's intervention and would not have been claimed otherwise. The dataset contains 4574 rescues with 749 negative ones among which 672 were not claimed by anyone and 77 were claimed within the last hour of the pick up window by the selected group.

3.1 Feature engineering

We use a number of features for the prediction. The first group of features are directly related to the rescue, such as the travel time and distance between the donor and the recipient generated by Google Maps Platform, the weight of the food, time of day, and which time slot the rescue belongs to.

We also used the weather information on the day of rescue from Climate Data Online, including the average temperature, precipitation and snowfall, as data analysis suggests that weather is correlated with the rescue outcome (Fig. 2a).

The third group of features involve the number of available volunteers near the donor and recipient's locations. Instead of using zip code, we evenly divide the area of operation of 412FR into a grid with 300 cells because the zip

Features	Rescue 1	Rescue 2
Fastest travel time of rescue	8 min	28 min
Travel distance of rescue	2.4 miles	18 miles
Weight of the food	5 lb	20 lb
Time of day	1pm	2pm
Time Slot	Weekday Afternoon	Weekend Afternoon
Precipitation	0	0.12 inch
Snowfall	0	0
Average temperature	62 $^\circ F$	76 $^\circ F$
AVs in donor's cell	20	91
Average AVs in donor's neighboring cells	40	250
AVs in recipient's cell	30	300
AVs in donor and recipient's cells with vehicle	21	116

Table 1: Two example data points for the predictive model.

code districts vary a lot in size (Fig. 2b) and the grid allows for better specificity. Each volunteer could set in their app the time slots they do not want to receive any notifications, which can also be interpreted as their availability. An *active* volunteer (AV) in a grid cell for a rescue is one who had done a rescue in the cell and marked themselves as available for the rescue's pick-up time slot in the app. We use as feature the number of AVs in the donor's and recipient's cells and the number of them averaged over the cells adjacent to the donor's. The number of AVs who indicate they have vehicles is helpful as well, as those without vehicles might be more constrained in their choice of rescues.

We also tested some other features such as average household income and vehicles. However, they do not improve the performance of the model. An example of the features we use for the training the machine learning model is shown in Table 1. These two data points are for illustration purpose and are not real rescues, as per our agreement with 412FR.

3.2 Stacking Model

We first attempted a few baseline models including Gaussian Process (GP) and Random Forest (RF) with different parameters but got unsatisfying performance, especially with the false positives, i.e. when the rescue is unclaimed but we predict it as claimed. In the context of food rescue, we want to inform human dispatchers which rescues will be unclaimed without human intervention and need extra attention. Thus, false positives can be costly because it may lead to the ignorance of a rescue in need of intervention and the waste of donated food, while false negatives are less concerning because it only leads to unnecessary extra attention from the human dispatcher. To deal with weak learners, we use a stacking approach inspired by (Wolpert 1992), whose structure is shown in Fig. 3. First, we split the training data into two sets, D_A and D_B. We use D_A to train various base models (Fig. 3, ①) and then we use these trained models to make predictions on D_B (Fig. 3, ②). Finally, we train a meta learner using the base models' predictions on D_B to determine the stacking model's estimate (Fig. 3, ③). In our case, we use 5 GP regressors and 1 RF classifier as the base model. The 5 GPs have different kernels and parameters for length scales. The parameters for GPs are shown in Table 2. The Random Forest Classifier has 100 estimators and the max-depth for any decision tree is 9.

GP	1	2	3	4	5
Kernel	DP	DP	Matern	RBF	RBF
Alpha	0.5	0.01	0.3	0.1	0.03

Table 2: GP parameters. Alpha is the dual coefficient of training data points in kernel space. DP means dot product.

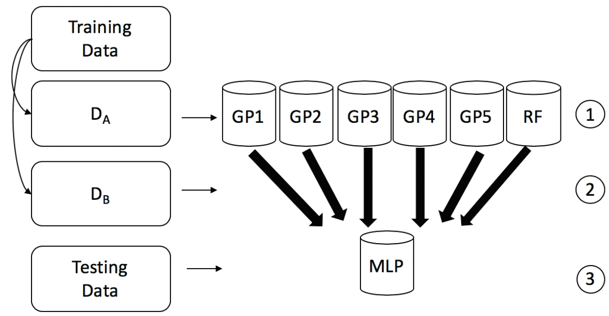


Figure 3: The stacking model.

All the six models are trained on the same data D_A. We use the mean values of the GPs' predictions and the binary label of the RF classifier, on D_B, as the input to the neural network meta learner. We report the results in Sec. 5.

4 Optimizing Intervention and Notification

We also perform prescriptive analysis to optimize the INS of 412FR, determining the guideline for dispatcher intervention and the rules for sending notifications. Our goal is to reduce the frequency that the dispatcher intervenes to "save" a rescue, or the mobile app notifications sent, ideally both.

We formalize the problem by defining an INS as a tuple (x, y, z), with x, y, z described below. When a rescue is posted, the mobile app first sends notifications to the volunteers who are within y miles from the donor. If no volunteer claims the rescue within the first x minutes, the app then sends the notification again to all volunteers who have indicated availability in the corresponding time slot. The dispatcher monitors the rescue after it is posted. If a rescue has not been claimed by z minutes before its pickup deadline, the dispatcher intervenes by directly contacting a group of regular volunteers and asking them if they are willing to claim it. If w_r is the duration from the posting time to the pickup deadline of rescue r, then the dispatcher intervenes $w_r - z$ minutes after the rescue is posted. We assume that upon the dispatcher's intervention, with probability μ the rescue immediately gets claimed, otherwise it has no effect.

412FR has always used a default INS: $\hat{x} = 15$ (minutes), $\hat{y} = 5$ (miles), and $\hat{z} = 60$ (minutes). We look for the optimal INS in a finite set S of candidate INSs which minimizes

$$\lambda \mathbb{E}_{r \sim R}[c_1(x, y, z, r)] + \mathbb{E}_{r \sim R}[c_2(x, y, z, r)] \quad (1)$$

Notation	Meaning
x	Second round notification time, default $\hat{x}$
y	First round notification radius, default $\hat{y}$
z	Intervention time from deadline, default $\hat{z}$
μ	Dispatcher intervention success probability
r	r.v.: a rescue, following distribution R.
w_r	Duration from r being posted to deadline
λ	Trade-off of intervention and notification.
$s(\cdot)$	Average number of dispatcher interventions
$v(\cdot)$	Average number of 1st round notifications
$q(\cdot)$	Average number of 2nd round notifications
$p(a, \cdot)$	Proportion of rescues claimed in a minutes
S	Domain of optimization variables (x, y, z)
b_i	Claim rate lower bound

Table 3: Notations for the optimization problem.

where R is the distribution of rescues, and λ controls the trade-off between two quantities: the expected number of dispatcher interventions and the expected number of notifications sent to volunteers. c_1 is the average number of dispatcher intervention for rescue r, c_2 is the average number of notifications sent for r given the INS. We also want to maintain a high claim rate, i.e., $E_{r\sim R}[c_3(a_i, x, y, z, r)] \geq b_i$ for a given set of a_i, b_i where c_3 is the probability that rescue r is claimed within first a_i minutes.

Without knowing the exact distribution R, we can only estimate these expected values through data. Given a dataset D of rescues under INS (x, y, z), we define $p(a, x, y, z)$ as the proportion of rescues in D that are claimed in a minutes; $s(x, y, z)$ as the proportion of rescues in D that are not claimed by volunteers before the dispatcher intervenes; $v(y)$ as the average number of available volunteers who are within y miles of the donor who receive the first round notifications; $q(x, y, z)$ as the average number of available volunteers who receive the second round notifications. Formally,

$$p(a, x, y, z) = \frac{1}{|D|} \sum\nolimits_{r \in D} \mathbb{I}\left(\text{rescue } r \text{ claimed in } a \text{ min}\right),$$

$$s(x, y, z) = \frac{1}{|D|} \sum\nolimits_{r \in D} \mathbb{I}\left(r \text{ not claimed in } w_r - z \text{ min}\right),$$

$$v(y) = \frac{1}{|D|} \sum_{r \in D} \text{\# available volunteers within } y \text{ miles of } r$$

$$q(x, y, z) = \frac{1}{|D|} \sum_{r \in D} \mathbb{I}\begin{pmatrix} r \text{ not claimed} \\ \text{in } x \text{ min} \end{pmatrix} \times \begin{matrix} \text{\# available} \\ \text{volunteers for } r \end{matrix}$$

Assuming data points in D are sampled from R, we have

$$\mathbb{E}_{r\sim R}[c_1(x, y, z, r)] \approx s(x, y, z)$$
$$\mathbb{E}_{r\sim R}[c_2(x, y, z, r)] \approx v(y) + q(x, y, z)$$
$$\mathbb{E}_{r\sim R}[c_3(a, x, y, z, r)] \approx p(a, x, y, z)$$

Our final optimization problem is as follows.

$$\min_{x,y,z} \quad C(x, y, z) = \lambda s(x, y, z) + v(y) + q(x, y, z) \tag{2}$$

$$s.t. \quad p(a_i, x, y, z) \geq b_i, \quad \forall i \in I \tag{3}$$
$$(x, y, z) \in S$$

From the historical data and dispatcher's advice, we could estimate μ, V_y, a_i, b_i, S. However, estimating $s(\cdot), q(\cdot), p(\cdot)$ poses significant difficulty. We need to estimate the counterfactual claim time (CCT) for all INSs $(x, y, z) \neq (\hat{x}, \hat{y}, \hat{z})$.

4.1 Counterfactual claim time (CCT) estimation

Given a rescue happened under the default INS $(\hat{x}, \hat{y}, \hat{z})$, we estimate its CCT under some other INS (x, y, z). We make the following assumptions.

- No matter when a volunteer receives the notification, upon receiving it they take the same amount of time to respond, and the effect of human intervention is independent of the app notification.
- The intervention outcome is not affected by the INS.
- Given a list of regular volunteers (provided by dispatchers or derived from data), if a rescue is recorded in the historical data as claimed by a regular volunteer after the dispatcher intervention time, i.e., $w - \hat{z}$ minutes after the rescue is posted, we give the credit to dispatcher intervention. If a rescue was claimed after the dispatcher intervention time by anyone else, we assume that the dispatcher's intervention have failed.

Suppose the rescue was claimed by volunteer i located d miles from the donor in the historical data. At a high level, in most cases we compute the claim time of volunteer i in the new INS (x, y, z) and take that as our CCT estimate. For example, suppose i is within the first round notification radius, i.e. $d \leq \hat{y}$ and claims the rescue in 7 minutes under $(\hat{x}, \hat{y}, \hat{z})$. This rescue would have a CCT of 12 minutes when $x = 5, z = \hat{z} = 60$ and $y < d \leq \hat{y}$, i.e., i is now outside the first round notification radius. This is because the volunteer i needs 7 minutes to respond after getting notification, but now they only receive the notification 5 minutes after the rescue is available. We also factor in the effect of dispatcher intervention when the intervention happens before the CCT k, i.e. $w_r - z < k$. For rescue r, we report the expected claim time $m_z(k) = \mu \min\{w_r - z, k\} + (1 - \mu)k$. In another scenario, if in the historical data, volunteer i who is not in the first round notification radius claims the rescue before the second round notification, we assume the volunteer's action is due to actively checking the available rescues and is not affected by the notification. Thus, the CCT remains the same for all INS. The complete computation is shown in Fig. 4.

We claim that our estimation is conservative, i.e., we will never underestimate the claim time. This is important in practice, because overestimation may merely lead to unnecessary resource spent but underestimation may cause a rescue to fail. Our estimation is accurate when i is within the first round notification radius in the counterfactual INS but not in the default INS and intervention happens after the claim time, as i would still be the first volunteer to claim the rescue under the counterfactual INS. In some other cases, there exists the unobservable possibility that some other volunteer might claim the rescue before i in the counterfactual INS, and hence we might overestimate the claim time.

Dispatcher intervention	Notification radius	Distance	Counterfactual claim time
$a < w_r - \hat{z}$	$y \leq \hat{y}$	$d \leq y$	$m_z(a)$
		$d \in (y, \hat{y}]$	$m_z(a + x)$
		$d > \hat{y}$	$m_z(a + x - \hat{x})$
	$y > \hat{y}$	$d \leq y$	$m_z(a)$
		$d \in (\hat{y}, y]$	$m_z(a - \hat{x})$
		$d > y$	$m_z(a + x - \hat{x})$
$a \geq w_r - \hat{z}$, by regular volunteer	any	any	z
$a \geq w_r - \hat{z}$, by other volunteer	$y \leq \hat{y}$	$d \leq y$	a
		$d \in (y, \hat{y}]$	$a + x$
		$d > \hat{y}$	$a + x - \hat{x}$
	$y > \hat{y}$	$d \leq y$	a
		$d \in (\hat{y}, y]$	$a - \hat{x}$
		$d > y$	$a + x - \hat{x}$

Figure 4: Construction of the CCT for INS (x, y, z) based on default INS $(\hat{x}, \hat{y}, \hat{z})$. a is the rescue's actual claim time. d is the distance from the rescue's volunteer to the donor.

4.2 Solving the optimization problem

Given the CCT estimate for each rescue, we can estimate the functions $s(\cdot), q(\cdot), p(\cdot)$ using the counterfactual dataset. However, there is no closed-form expression for them. Computing their values at every point in a brute force way is obviously inefficient. We propose a branch-and-bound algorithm and a feasibility check to find optimal INS more efficiently.

First, we note that the CCT, as detailed in Fig. 4, is increasing in x and decreasing in y and z. Since $p(\cdot)$ is the empirical estimate based on the claim time, if some infeasible INS (x, y, z) does not satisfy claim rate constraint (3), any INS $(\tilde{x}, \tilde{y}, \tilde{z})$ with $\tilde{x} \geq x, \tilde{y} \leq y, \tilde{z} \leq z$ is also infeasible. Thus, we need not generate CCT for $(\tilde{x}, \tilde{y}, \tilde{z})$.

Using a similar observation, we devise our main algorithm, Alg. 2. Note that $s(x, y, z)$ decreases as x, z decreases and y increases, $v(y)$ decreases as y decreases, $q(x, y, z)$ decreases as x, y, z increases. Therefore, if we replace all the variables in all terms with the extreme values in domain S that can minimize $C(x, y, z)$ (as shown in Table 4), we get a lower bound of $C(x, y, z)$. We define a subproblem as the original optimization problem with k of the variables in the INS specified and the remaining ones unspecified for $k = 0, 1, 2, 3$. To compute a lower bound for each subproblem, we replace the unspecified variables in each term with the extreme values according to Table 4. For example, if z is specified, and x, y are unspecified, we get a lower bound

$$\bar{C} = \lambda s(x_{min}, y_{max}, z) + v(y_{min}) + q(x_{max}, y_{max}, z)$$

In Alg. 2, we start with the original problem where none of the variables are specified ($k = 0$). We branch to lower level subproblems in the order of $z \to y \to x$, as this order tends to prune the fastest. For each subproblem, we either compute a lower bound, or when all variables are specified, compute the exact cost. We generate one counterfactual dataset for computing the exact cost (Line 3, Alg. 1), and at most two datasets when computing the lower bound (Line 8, Alg. 1), since $s(\cdot)$ and $z(\cdot)$ are minimized at two different INSs and $v(\cdot)$ does not depend on the CCT. The implicit pruning on Line 3 guarantees Alg. 2 finds the optimal solution.

$s(x, y, z)$	x_{min}	y_{max}	z_{min}
$v(y)$		y_{min}	
$q(x, y, z)$	x_{max}	y_{max}	z_{max}

Table 4: Replace (unspecified) variables in each term with the extreme values to get a lower bound.

Algorithm 1: Solve-Relaxation

1 Optional input arguments: x, y, z
2 **if** *all of* x, y, z *specified* **then**
3   Generate CCTs with (x, y, z).
4   **if** *feasible* **then**
5   Compute cost $\bar{C} = C(x, y, z)$
6   **return** subproblem $(\bar{C}, (x, y, z))$
7 **else**
8   Generate CCTs with unspecified parameter replaced by extreme values in Table 4.
9   Compute lower bound $\bar{C}$
10   **return** subproblem $(\bar{C}, (x, y, z))$

5 Results

5.1 Prediction

We "predict the future with the past". As mentioned in Section 3, we treat rescues done by volunteers who have done over 10 rescues in the last two months in the last hour of the pick-up window as unclaimed. Thus, we exclude the rescues in the first two months from our prediction task as we do not have the volunteer history for these early entries. As a result, the training data consist of rescues from May 2018 to December 2018 and testing data consist of rescues from January 2019 to May 2019. In addition, since the dataset is imbalanced on the number of claimed and unclaimed rescues, we oversample the unclaimed rescues so that the ratio of claimed and unclaimed rescues is 1:1. The oversampling is applied to only the training dataset for the predictive model.

Table 5 shows the stacking model outperforms all baseline models. Moreover, it yields almost no false positive errors. This is especially important in the food rescue operation, as the cost of not taking actions to a rescue which turns out unclaimed due to a false positive is much higher than that of an unnecessary dispatcher intervention due to a false negative.

5.2 Optimization

After consulting the dispatcher, we take $\mu = 0.4$ as the probability that dispatcher intervention is effective. We require that the optimal INS's claim rate be no worse than default INS. That is, we use $a_i = 1, 2, \ldots 120$ and b_i being the empirical claim rate at the a_i-th minute under the default INS.

First, we demonstrate the effectiveness of the branch and bound algorithm (Alg. 2). We set the domain S as $x, y \in \{2, 4, 6, 8\}$ and $z \in \{30, 40, 50, 60\}$. As shown in Table 6, branch and bound needs to generate CCTs on much less INSs than the brute force approach, although advantage is less significant for smaller λ. In the sequel, we use

Algorithm 2: Branch-and-Bound

Push Solve-Relaxation($\{\}$) to Frontier.
while *Frontier set is not empty* **do**
  Get subproblem with lowest $\bar{C}$ from Frontier.
  if *subproblem has all parameters specified* **then**
    return $(\bar{C}, (x, y, z))$ #(optimal solution)
  else
    Follow the order $z \rightarrow y \rightarrow x$ to expand the node, i.e., if the first k variables are already specified, create a subproblem for each possible value of the $(k+1)^{th}$ variable in S.
    Add all subproblems Solve-Relaxation(x, y, z) to Frontier.

Model	Accuracy	Precision	Recall	F1	AUC
GB	0.73	0.86	0.82	0.84	0.51
RF	0.71	0.87	0.78	0.82	0.54
GP	0.56	0.88	0.54	0.67	0.60
SM	0.69	1.00*	0.64	0.78	0.81

Table 5: Performance of selected models, GB: Gradient Boosting Classifier, RF: Random Forest; GP: Gaussian Process; SM: Stacking Model. * We run the experiments for SM for 3 times. The precisions are 1.0, 1.0, 0.9969.

Alg. 2 and set the domain S as $x \in \{1, 1.5, 2, \ldots, 25\}$, $y \in \{1, 1.5, 2, \ldots, 10\}$, $z \in \{30, 32.5, 35, \ldots, 90\}$.

Similar as above, we use the earlier data D_{past} to predict the more recent data D_{future}. First, we focus on computing the optimal INS on D_{past} in Fig. 6a. Both the 2nd round notification time and the dispatcher intervention time decrease as λ grows, i.e. the dispatcher's intervention matters more in the dispatching cost. This is aligned with the results in Table 4. When the app notification is the primary concern, the default INS is almost desirable, yet if we would like to minimize the interventions, the 2nd round notification needs to go out sooner. Fig. 6b, we show the Pareto frontier (in red) of optimizing on D_{past}. The optimal INSs in Fig. 6a are now shown in blue. The default INS lies within the frontier, suggesting that the numbers of both interventions and notifications can be improved. The orange rectangle indicates the INS region that is strictly superior to the default INS.

Of course, we would like to examine the quality of the optimization solutions on unseen data. Thus, in Fig. 6c, we show the projected number of interventions and notifications on D_{future} of the optimal INSs on D_{past}. For the same INS, the performance is different between Fig. 6b and Fig. 6c because the claim probabilities are estimated using the two datasets separately. Despite this difference, some optimal INSs on D_{past} still outperforms the current practice. We therefore suggest two INSs (see Fig. 6c) to 412 Food Rescue, as shown in Table 7. INS A is a strict improvement over the current practice, reducing the number of both intervention and notification. Our second solution, INS B, drastically reduce the labor of dispatcher by 24% at the expense of a mere 2% increase of notifications sent. Since 412FR handles 4574

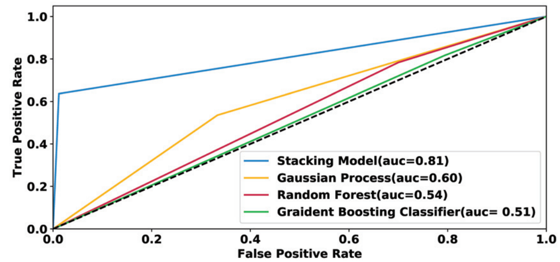


Figure 5: The ROC curves of the models

	Brute force search		Branch and bound	
λ	INSs	Time (s)	INSs	Time (s)
10^7	64	192.6	18	65.5
10^6	64	183.7	18	64.9
10^5	64	185.1	16	56.4
10^4	64	190.9	34	125.0
10^3	64	187.1	35	129.3

Table 6: Running time and the number of INSs for which the CCTs are generated.

rescues in a 430-day period, INS B can save the dispatcher over 390 times of intervention a year in expectation. An intervention takes the dispatcher at least the same amount of time as matching a new food rescue, and often more. Thus, the dispatcher could handle at least 390 extra rescues a year, which is over 7500 pounds of food by the average donation in our dataset. We choose INS B over the rightmost INS on Fig. 6c because 412FR has relatively more shortage of dispatcher than volunteers. Finally, Fig. 6d shows our two INSs have competitive claim rates on the unseen data. This suggests the promise of deploying our method in the future.

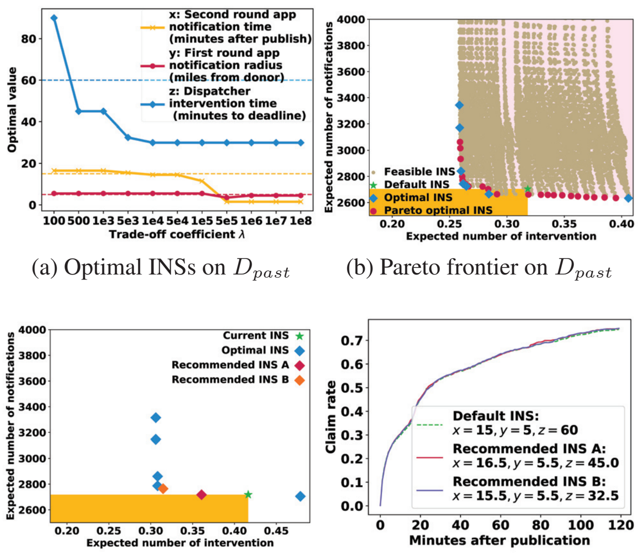
6 Discussion

As mentioned in Section 2, there are other facets of the food rescue problem that can be tackled with a computational approach, including optimizing the matching between donors and recipients, and directly incentivizing volunteers to be more active. We focus on predicting whether a rescue can be claimed and optimizing INS as they are of higher priority to our partners at 412FR with a clearer path for future deployment. Specifically, the matching is currently done manually at 412FR. Rather than a mechanical process, the dispatcher's job requires a high level of situational judgment, interpersonal skills, and the rapport developed over time. Through our multiple conversations with 412FR and direct experience of shadowing the dispatcher, we believe that currently, automating the matching would not benefit 412FR without a big change of the overall workflow involving all the donors, recipients as well as 412FR. Motivating the volunteers to boost claim rate would require 412FR to take new initiative that is not in place. In contrast, the INS is a current practice and it is easier to test our solution. Nonetheless, we will consider these directions as we continue to work with 412FR.

In addition, our optimization framework is already tak-

INS	Interventions	Notifications
A: $(16.5, 5.5, 45)$	$-13\%\,(-0.06)$	$0\%\,(-1)$
B: $(15.5, 5.5, 32.5)$	$-24\%\,(-0.10)$	$+2\%\,(+46)$

Table 7: The projected change in the probability of interventions and number of notifications of the proposed INSs. The numbers in parentheses are the absolute change.



(a) Optimal INSs on D_{past}

(b) Pareto frontier on D_{past}

(c) Performance of the optimal INSs for D_{past} on D_{future}

(d) Projected claim rate of two recommended INSs on D_{future}

Figure 6: Experiment results of the data-driven optimization

ing into account the volunteer retention problem implicitly as we try to reduce the notifications sent in the objective function. Our framework can be extended to different objectives, and one may design an objective that explicitly focuses on volunteer retention. For example, a user's probability of uninstalling an app could be modeled as a function $f(\cdot)$ of the number of notifications they receive in a week (Gibb 2018). To minimize the number of volunteers lost, we could change the optimization objective from Eq. (1) to

$$\mathbb{E}_{n\sim N}\mathbb{E}_{r_1,\ldots,r_n\sim R^n}\sum\nolimits_{i\in V}\mathbb{E}_{n_i\sim N_i(\{r_j\},x,y,z)}f(n_i)$$

where n is the number of total rescues in a week following some distribution N, V is the set of volunteers, and n_i is the number of notifications volunteer i receives in a week following a distribution determined by the set of rescues and the INS. Similar to Eq. (1), this objective value can be estimated through the counterfactual datasets.

Another promising direction is to combine our prediction model and the optimization algorithm to optimize rescue-specific INS, for which our Alg. 2 can be adapted to work. We defer the further investigation to future work.

7 Conclusion

We provide the first predictive and prescriptive analysis of volunteer-based food rescue operations. Our stacking model predicts the claim status of rescues with AUC of 0.81. Such prediction helps the dispatcher better prepare for interventions and alleviate their uncertainty. Our data-driven optimization reduces the frequency of dispatcher intervention and push notifications sent to volunteers, without harming the claim rate. The dispatcher can use the saved effort to handle an extra 7500 pounds of food a year that would otherwise go to waste. By improving the operation efficiency of inspiring organizations like 412FR, our research contribute to the fight against food waste and insecurity. We are working with 412FR to deploy our results in the near future.

Acknowledgments

This work was supported in part by NSF grant IIS-1850477. We sincerely thank our partners at 412 Food Rescue.

References

Coleman-Jensen, A.; Rabbitt, M. P.; Gregory, C. A.; and Singh, A. 2018. Household food security in the united states in 2017. *USDA-ERS Economic Research Report*.

Conrad, Z.; Niles, M. T.; Neher, D. A.; Roy, E. D.; Tichenor, N. E.; and Jahns, L. 2018. Relationship between food waste, diet quality, and environmental sustainability. *PloS one* 13(4):e0195405.

Dehejia, R. H., and Wahba, S. 2002. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*.

Felt, A. P.; Egelman, S.; and Wagner, D. 2012. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, 33–44. ACM.

Gibb, R. 2018. How consumers perceive push notification in 2018.

Gunes, C.; van Hoeve, W.-J.; and Tayur, S. 2010. Vehicle routing for food rescue programs: A comparison of different approaches. In *CPAIOR*.

Ho, C.-J., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *AAAI*.

Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *STOC*.

Lee, M. K.; Kusbit, D.; Kahng, A.; Kim, J. T.; Yuan, X.; Chan, A.; See, D.; Noothigattu, R.; Lee, S.; Psomas, A.; and Procaccia, A. D. 2019. Webuildai: Participatory framework for algorithmic governance. *Proc. ACM Hum.-Comput. Interact.* 3(CSCW).

Lundy, T.; Wei, A.; Fu, H.; Kominers, S. D.; and Leyton-Brown, K. 2019. Allocation for social good: auditing mechanisms for utility maximization. In *ACM EC*.

Mehta, A.; Waggoner, B.; and Zadimoghaddam, M. 2015. Online stochastic matching with unequal probabilities. In *Proceedings of the twenty-sixth ACM-SIAM symposium on Discrete algorithms*.

Nair, D.; Grzybowska, H.; Fu, Y.; and Dixit, V. 2018. Scheduling and routing models for food rescue and delivery operations. *Socio-Economic Planning Sciences* 63:18–32.

Phillips, C.; Hoenigman, R.; and Higbee, B. 2011. Food redistribution as optimization. *arXiv preprint arXiv:1108.5768*.

Pratt, C. C.; McGuigan, W. M.; and Katzev, A. R. 2000. Measuring program outcomes: Using retrospective pretest methodology. *American Journal of Evaluation* 21(3):341–349.

Prendergast, C. 2016. The allocation of food to food banks. *EAI Endorsed Trans. Serious Games* 3(10):e4.

Wolpert, D. 1992. Stacked generalization. *Neural networks*.