

## Article

# A Machine Learning Approach for Detecting Rescue Requests from Social Media

Zheye Wang <sup>1,\*</sup>, Nina S. N. Lam <sup>2</sup>, Mingxuan Sun <sup>3</sup>, Xiao Huang <sup>4</sup>, Jin Shang <sup>5</sup>, Lei Zou <sup>6</sup>, Yue Wu <sup>7</sup>  
and Volodymyr V. Mihunov <sup>2</sup>

<sup>1</sup> Kinder Institute for Urban Research, Rice University, Houston, TX 77005, USA

<sup>2</sup> Department of Environmental Sciences, Louisiana State University, Baton Rouge, LA 70808, USA

<sup>3</sup> Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA 70803, USA

<sup>4</sup> Department of Geosciences, University of Arkansas, Fayetteville, AK 72762, USA

<sup>5</sup> Amazon, Seattle, WA 98109, USA

<sup>6</sup> Department of Geography, Texas A&M University, College Station, TX 77843, USA

<sup>7</sup> Department of Geography & Anthropology, Louisiana State University, Baton Rouge, LA 70802, USA

\* Correspondence: zw60@rice.edu

**Abstract:** Hurricane Harvey in 2017 marked an important transition where many disaster victims used social media rather than the overloaded 911 system to seek rescue. This article presents a machine-learning-based detector of rescue requests from Harvey-related Twitter messages, which differentiates itself from existing ones by accounting for the potential impacts of ZIP codes on both the preparation of training samples and the performance of different machine learning models. We investigate how the outcomes of our ZIP code filtering differ from those of a recent, comparable study in terms of generating training data for machine learning models. Following this, experiments are conducted to test how the existence of ZIP codes would affect the performance of machine learning models by simulating different percentages of ZIP-code-tagged positive samples. The findings show that (1) all machine learning classifiers except K-nearest neighbors and Naïve Bayes achieve state-of-the-art performance in detecting rescue requests from social media; (2) using ZIP code filtering could increase the effectiveness of gathering rescue requests for training machine learning models; (3) machine learning models are better able to identify rescue requests that are associated with ZIP codes. We thereby encourage every rescue-seeking victim to include ZIP codes when posting messages on social media. This study is a useful addition to the literature and can be helpful for first responders to rescue disaster victims more efficiently.

**Keywords:** rescue request; twitter; machine learning; Hurricane Harvey; deep learning



**Citation:** Wang, Z.; Lam, N.S.N.; Sun, M.; Huang, X.; Shang, J.; Zou, L.; Wu, Y.; Mihunov, V.V. A Machine Learning Approach for Detecting Rescue Requests from Social Media. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 570. <https://doi.org/10.3390/ijgi11110570>

Academic Editors: Wei Huang and Wolfgang Kainz

Received: 30 September 2022

Accepted: 11 November 2022

Published: 16 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Rescuing victims out of harm's way during disasters is a critical component of disaster response. Efficient rescue practices can save more lives on time. However, in rapid large-scale disastrous events such as hurricanes and flooding, many rescue requests often come at the same time, overloading the responding system, and making rescue efforts more difficult. For example, Hurricane Harvey struck the Houston area in 2017, resulting in massive flooding and urgent needs for rescue. There were too many calls to 911, crashing the call system and prompting residents to resort to social media and other means to request rescue [1,2].

Social media is reshaping our way of seeking rescue in disaster situations. The emergence of social networking sites such as Twitter and Facebook have enabled people to not only share their personal life but also to post rescue requests and communicate with disaster responders [1,3–8]. These social media messages from the public can be valuable data for improving disaster response and resilience [5]. However, as a type of user-generated content (UGC), social media messages are unstructured and often contain

noises and non-informative content. This has imposed pressure on emergency responders to filter enormous numbers of social media messages [9]. In spite of this, researchers have attempted to extract help and rescue information from social media.

These efforts often involve training machine learning models to achieve automatic information extraction [10–13]. Models used in these studies include Decision Tree [11], Multilayer Perceptron [11], K-Nearest Neighbors [14,15], Classification and Regression Trees (CART) [15], Naïve Bayes [10,13,14], Logistic Regression [11,12,14,15], Random Forest [10], Support Vector Machine [10–12,15], Convolutional Neural Network [11,12,16], Long Short-Term Memory [12], and Bidirectional Encoder Representations from Transformers (BERT) [16]. Similar to most machine learning applications in natural language processing, their workflows generally follow the following three major steps:

- (1) Extracting a subset of Hurricane Harvey tweets and manually labeling them based on a classification schema.
- (2) Training machine learning models with a certain percentage of this subset (training data) to build classifiers.
- (3) Evaluating model performance with test data.

Their major objective is often to train and test different models to best classify disaster posts based on a given classification schema. However, most of these schemas are too broad for first responders to directly differentiate rescue requests from a large number of other messages [10,13,14]. Several exceptions include Yang et al. [15], Devaraj, Murthy, and Dontula [11], Kabir and Madria [12], Zhou et al. [16]. The next section covers more details regarding these relevant studies. Different from these existing studies, the present research focuses on extracting rescue requests that are defined strictly as rescue-seeking messages with addresses, since only such tweets can be actionable for first responders. Intuitively, addresses often include ZIP codes which may serve as a useful feature to separate rescue requests from other messages. However, few studies have examined the contribution of this feature to extracting rescue requests from a large volume of social media messages. The primary objectives of this article are two-fold: (1) investigating the potential benefits of ZIP code filtering in retrieving rescue requests for training models; (2) examining the impact of ZIP code presence on the performance of machine learning models in identifying rescue-seeking postings. This study could contribute to the literature by shedding more light on a specific feature in rescue requests that has been neglected by the majority of other investigations.

## 2. Related Work

We identify two major streams of relevant work in retrieving rescue and/or help information from social media data.

### 2.1. Classification of Disaster-Related Social Media Posts for Situational Awareness

To improve situational awareness in disaster situations, researchers have developed classifiers to extract useful and informative messages from disaster social media posts [5,10,13,14,17–19]. Some classifiers are binary. For example, Huang et al. [19] classified flooding tweets into two categories including “On-topic” and “Off-topic”, and Huang et al. [18] designed an extractor to automatically tag flood-related tweets. Multi-category classifiers have also been developed in an attempt to maximize the information extraction. de Albuquerque [17] classified a set of flooding tweets into several thematic groups including “Volunteer Actions”, “Media Reports”, “Traffic Conditions”, “First-hand Observations”, “Official Actions”, “Infrastructure Damage” and “Other”. Imran et al. [13] classified tweets related to Joplin 2011 Tornado into five broad categories including “Caution & Advice”, “Affected People”, “Infrastructure/Utilities”, “Needs & Donations”, and “Other”. The category of “Affected People” represents the “reports and/or questions about missing or found people”, which may best relate to our definition of rescue requests among other categories. Imran [10] further developed a labeling scheme where “Missing, trapped, or found people” is more relevant to rescue requests than the other eight

labels in the scheme. There also exists a fine-grained classification schema developed by Huang et al. [14] where Hurricane Sandy tweets were categorized into four main classes (i.e., preparedness, response, impact, recovery) and 47 subclasses. One subclass named “Rescue” under the “Response” class captures “rescues of disaster victims”. Although this labeling is very close to our definition of rescue requests, it is mainly useful for enabling situational awareness. These categories, in line with the discussion in the Introduction section, are too general to be particularly helpful to first responders.

## 2.2. Extraction of Urgent/Rescue Requests from Disaster Posts

Few studies have extracted urgent/rescue requests that are directly useful for rescue efforts. Only several exceptions are found in the literature, including Yang et al. [15], Devaraj, Murthy, and Dontula [11], Kabir and Madria [12], Zhou et al. [16]. Yang et al. [15] trained and compared multiple machine learning models based on 1000 manually annotated tweets. Their best model came from SVM, which yielded an F score of 68.7%. Kabir and Madria [12] trained several machine learning models to distinguish “Rescue Needed” from other categories such as “Water Needed”, “Injured”, “Sick”, and “Flood”. Their best model came from an enhanced CNN model with an F score of 87.2%. Devaraj, Murthy, and Dontula [11] developed several binary classifiers using machine leaning to differentiate “Urgent Requests” from other disaster messages. Their best models (i.e., Support Vector Machine (SVM) and Convolutional Neural Network (CNN) classifiers) received the same F1 score of 87%. Zhou et al. [16] designed several classifiers for collection of rescue request tweets using pretrained language models, where the best model (a Bidirectional Encoder Representations from Transformers (BERT)-based model with a CNN classifier) achieved an F score of 91.9%.

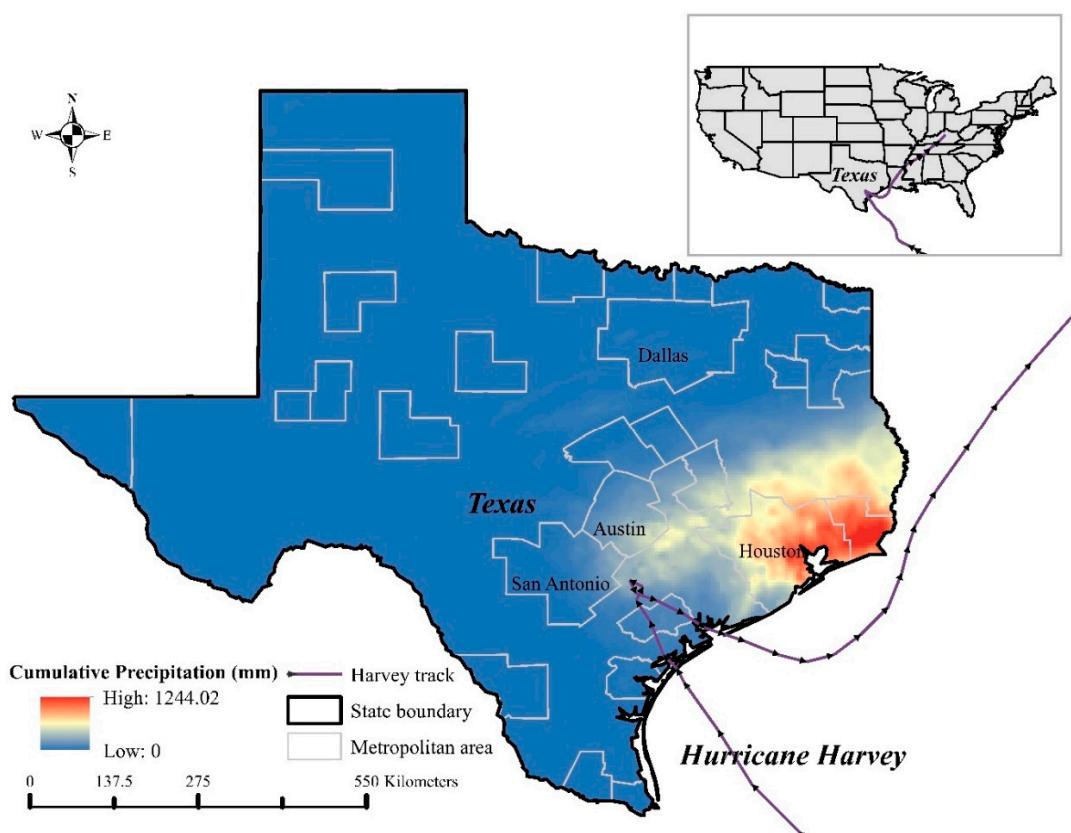
Under emergencies, disaster victims compose rescue request tweets differently, making it challenging to identify keywords that can best capture rescue request messages from the big noisy data generated on social media [1]. The aforementioned studies have shown that urgent/rescue requests using social media are relatively rare, making it difficult to obtain sufficient training samples. For example, Devaraj, Murthy, and Dontula [11] found that urgent requests in Hurricane Harvey are “uncommon” and that it is very challenging to find positive cases to prepare for the training data. They used rescue-related keywords including “help”, “rescue”, and “911” to filter urgent requests from Harvey tweets. However, these keywords do not efficiently separate rescue requests from other messages because not every message containing these keywords is rescue request. Their study actually reported that more than 90% of the tweets filtered by these keywords are not rescue-seeking messages.

Thus, this study addresses the question: are there alternative ways to filter rescue requests from disaster posts? More specifically, do rescue requesting posts share a specific feature that can distinguish them from other disaster messages? Would machine learning models benefit from this feature in classifying rescue requests? Recall that rescue requests by our definition should have addresses. It is therefore worth investigating if any geographic features in these addresses would help. This study focuses on ZIP codes, because ZIP codes are often included in addresses and their structured nature could make them easy to be identified. Developing algorithms that can seamlessly incorporate this geographic feature with machine learning methods has the potential to make breakthroughs in accurately detecting rescue requests on social media.

## 3. Data and Models

### 3.1. Collection of Hurricane Harvey Tweets

Developed from a tropical wave, Hurricane Harvey reached the tropical storm status on 17 August 2017. Eight days later, Harvey strengthened into a Category 4 hurricane and made its first landfall on San Jose Island, Texas. Harvey’s second landfall took place on 26 August morning near the northeast of Copano Bay, Texas. After downgrading to a tropical storm as it stalled near the coastline of Texas, Harvey brought a huge amount of rainfall to the Houston metropolitan area (Figure 1).



**Figure 1.** Study area, Hurricane Harvey track, and cumulative precipitation (27–31 August 2017).

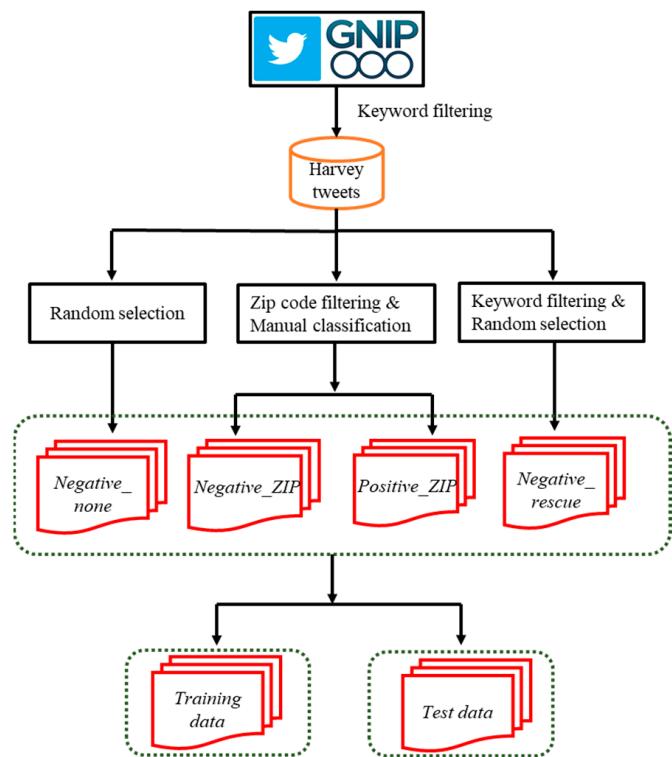
Hurricane Harvey tweets posted between 17 August 2017 and 7 September 2017 were purchased from Gnip (<https://support.gnip.com/sources/twitter/>, accessed on 10 May 2018) based on a keyword search procedure. That is, tweets containing any of the following keywords were gleaned and stored as JSON files:

*hurricane, harvey, disaster, cajun navy, hurricaneharvey, txdps, txtf1, redcross, coast-guard, houstonpolice, houstonoem, salvationarmy, flood, sos, flooding, storm, rescue, sendhelp, cajunnavy, fema, salvation army*

This resulted in 45 million tweets. The time span of the tweets collected was according to the three phases of disaster management, including preparedness, response, and recovery [2].

### 3.2. Training and Test Data

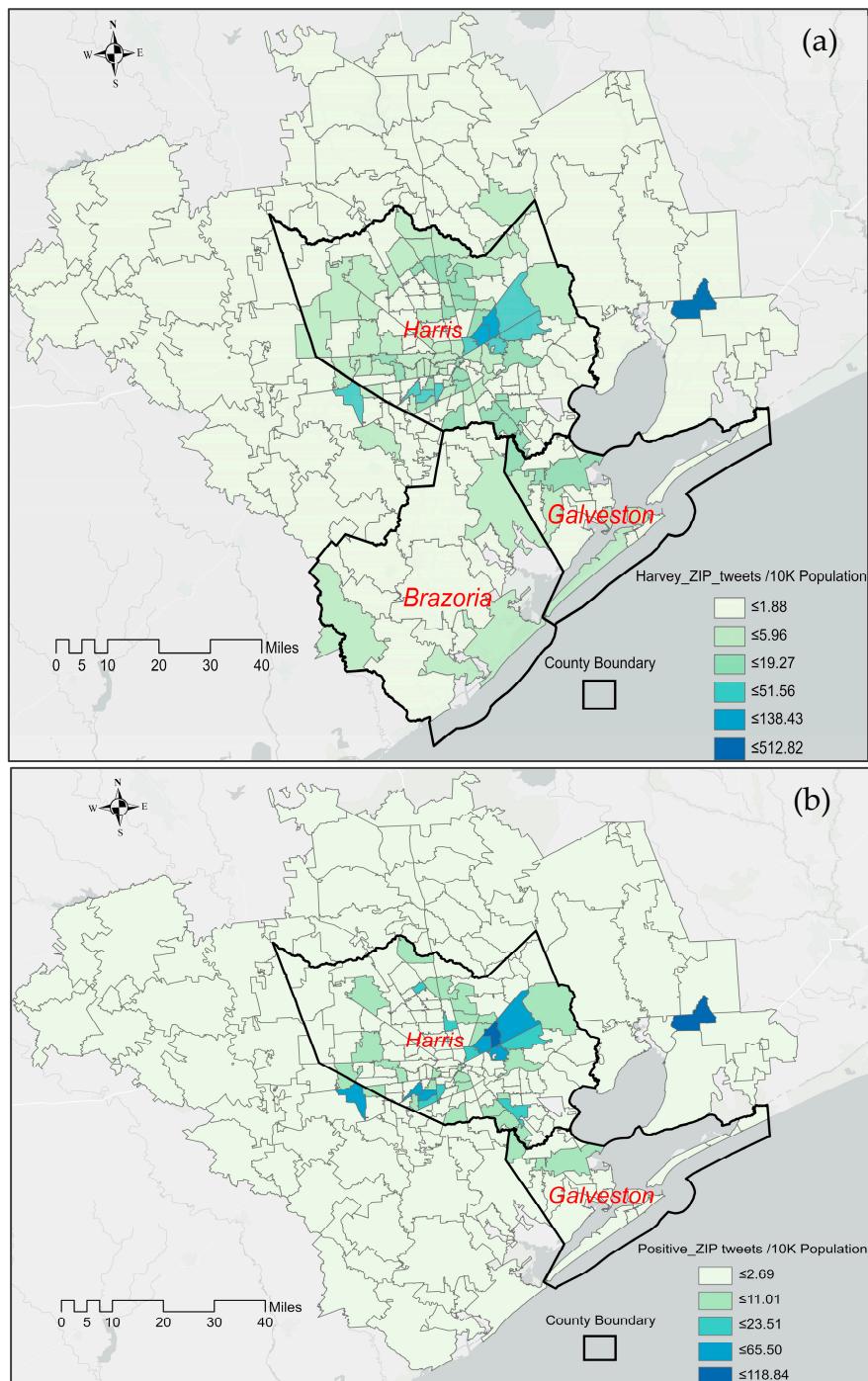
We used Harvey tweets posted between 27 August 2017 and 31 August 2017 (4.1 million tweets) for the training and test data selection. The workflow for the preparation of training and test data is shown in Figure 2. Note that retweets are not included in this dataset because they are duplicate messages of their original tweets. Generally, three types of geographic information can be found in Twitter messages, including profile locations, GPS (global positioning systems) locations, and locations in textual content [4]. Profile locations are self-reported locations (e.g., states, cities, and counties) from Twitter users, which do not contain fine-scale information for rescue efforts. GPS locations can only be obtained when devices had the built-in GPS turned on, resulting in the rarity of these geotagged tweets [3]. Given the quality of profile locations and the rarity of GPS locations, first responders often rely on the locations in Twitter textual content (i.e., geographic addresses in rescue requests) to geolocate disaster victims.



**Figure 2.** Workflow for preparing training and test data for machine learning models of rescue requests.

As mentioned in the Introduction section, a ZIP code filtering could be implemented to extract Harvey tweets containing addresses, given that ZIP codes are often included in them. The ZIP code filtering is also a keyword search procedure, whereas its keywords are all numbers (ZIP codes). A list of ZIP codes for the Houston Metropolitan Statistical Area (MSA) was complied. Harvey tweets containing any ZIP code in this list were obtained to build a dataset named as *Harvey\_ZIP\_tweets*, which consists of 2804 tweets. These tweets were tabulated and mapped according to their ZIP codes in Figure 3a. The map shows that these tweets cluster in three counties, where Harris is the central county of Houston MSA and Brazoria and Galveston are two coastal counties.

**Positive cases.** The positive cases were obtained by a manual classification of *Harvey\_ZIP\_tweets*. This binary classification resulted in a positive class (rescue requests) named as *positive\_ZIP* and a negative class named as *negative\_ZIP*. A total of 2106 *positive\_ZIP* tweets (rescue requests) were obtained through this step. That is, 75% of the Harvey tweets filtered by ZIP codes are rescue requests. This differs from the work of Devaraj, Murthy, and Dontula [11], where less than 8% of their tweets filtered with rescue-related keywords such as “help”, “rescue”, and “911” were labeled as urgent requests in Hurricane Harvey. This significant difference could be attributed to the distinctive filtering methods—our ZIP code filtering avoids the “needle in a haystack” situation. We also observe that some other keywords such as “stuck” and “trapped” in rescue requests were overlooked by Devaraj, Murthy, and Dontula [11]. This may also, to some extent, explain their difficulty in obtaining larger training data. The positive cases were also mapped based on their ZIP codes in Figure 3b, where two clusters including Harris and Galveston counties are found. These addresses coming with rescue requests can help locate disaster victims and schedule rescue efforts, while maps in Figure 3 are useful for situational awareness and emergency resource allocation.



**Figure 3.** Choropleth maps showing the geospatial distribution of *Harvey\_ZIP\_tweets* (a) and *positive\_ZIP* (b).

**Negative cases.** A negative class *negative\_ZIP* was yielded by the above binary classification. Yet, comprehensive negative samples should consist of the following three types: (1) tweets that contain ZIP codes but are not rescue-related, i.e., *negative\_ZIP*, (2) tweets that do not contain ZIP codes but are rescue-related, i.e., *negative\_rescue*, (3) tweets that neither contain ZIP codes nor relate to rescue, i.e., *negative\_none*. Note that the first type of negative samples was retained to confront the influence of the ZIP code feature so that machine learning would not solely rely on this single feature to differentiate rescue requests from other tweets. To obtain the second type of negative samples, we randomly selected 2000 tweets containing either 'rescue' or 'help' from the entire Harvey tweets and screened

them to assure that they were not rescue requests. We also randomly selected 1302 tweets from the Harvey tweets to obtain the third type of negative samples. In this way, a total of 4000 negative samples were obtained.

**Train–test split.** Finally, the whole dataset (both negative/positive, 6106 tweets) was split into training data (80%) and test data (20%).

### 3.3. Prediction Data

After removing the training data and test data, we used the remaining Harvey tweets on 28 August as the prediction data. These data consist of 995,732 Harvey tweets. Our trained classifier was then used to classify the prediction data and identify rescue requests accordingly.

### 3.4. Text Cleaning and Transformation

Twitter messages often contain words that are meaningless to our analysis. Therefore, a text cleaning is necessary to help filter out these words and obtain a clean corpus. Tweets in both the training data and the test data were cleaned with the following steps:

- (1) Remove URLs.
- (2) Remove stop words.
- (3) Replace 5-digit numbers with a pseudo word ‘zcode’. This is to change all ZIP codes to the same feature.
- (4) Remove words with only 1 letter.
- (5) Remove punctuations.
- (6) Convert to lower cases.

The above steps except step 3 were standard text cleaning used in many existing studies [13,14]. Texts cannot be directly processed by some machine learning models unless being transformed into numeric forms. A commonly used method for text transformation is the Term Frequency-Inverse Document Frequency (TF-IDF) computed as:

$$tf - idf(t, d) = tf(t, d) \times idf(t)$$

where  $tf(t, d)$  represents the number of times a word  $t$  that appears in a certain tweet  $d$ , and  $idf(t)$  is given by  $idf(t) = \log \frac{1+n}{1+df(t)} + 1$ , with  $n$  representing the total number of tweets and  $df(t)$  denoting the number of tweets containing the word  $t$ .

### 3.5. Machine Learning Models

All machine learning algorithms discussed here are supervised machine learning. That is, a classifier is trained by mapping features (predictors) to a desired response class variable based on a given model [20]. In a binary classification scenario, features are mapped to a binary response (0/1). In our case of rescue request detection, we trained a classifier to categorize a Harvey tweet as either a rescue request or others by learning features in these tweets. Several commonly used machine learning models and two deep learning models were used to train a binary classifier. Note that the scikit-learn library in Python [21] was used to train and test these machine learning models, while a Python deep learning framework, i.e., Keras (<http://keras.io>, accessed on 5 August 2021) was used for the two deep learning models. Additionally, note that deep learning did not utilize the TF-IDF text transformation as described above. Considering that deep learning models often involve a large amount of hyperparameters, we opted to use settings that have been successful in previous research rather than having costly tuning from scratch.

#### 3.5.1. Logistic Regression

Due to its simplicity and high interpretability, logistic regression (LR) has been widely used in many fields for binary data classification. It was used here to model the probability of rescue requests. The training of logistic regression involves only a few hyperparameters such as solver, penalty, and regularization. Hyperparameter settings for this model can be found in Table A2.

### 3.5.2. K-Nearest Neighbors

K-nearest neighbors (kNN) is a simple supervised machine learning algorithm assuming that near things are more similar. kNN predicts or classifies a new Harvey tweet by finding the  $k$  closest training tweets to it. Closeness can be measured with distance metrics such as Euclidean distance, Manhattan distance, and Minkowski distance. kNN has limited hyperparameters to tune, whereas  $k$  referring to the number of closest neighbors is the most important one. Its hyperparameter settings can be found in Table A1.

### 3.5.3. Naïve Bayes

Naïve Bayes (NB) is a supervised learning algorithm building upon the Bayes theorem. The term “naïve” refers to an assumption that all features are completely independent of each other [22]. Few critical hyperparameters in NB need to be tuned.

### 3.5.4. Random Forest

Random forest (RF) is an ensemble learning model that fits a couple of decision tree classifiers. Decision tree mirrors the human decision-making process and can be visualized with a flowchart-like structure. A decision tree is grown by splitting or not splitting each decision node into two sub-nodes in a recursive manner (i.e., recursive partitioning). Attribute selection measures (ASM) such as information gain, Gini index, and gain ratio are often used as splitting criteria. Random forest is an ensemble of many decision trees where each tree is constructed based on a bootstrap sample of the training data. It aggregates (by averaging) the prediction of each decision tree to improve the predictive accuracy and control overfitting. Note that, when splitting a node to grow a decision tree, the partition is no longer based on all features. Instead, the best split is selected among a random subset of the features. The hyperparameters of random forest include the number of trees, the number of features to consider when looking for the best split, the maximum depth of the tree, the minimum number of samples required to split an internal node, and the minimum number of samples required to be at a leaf node, and others. Table A3 reports its hyperparameter settings.

### 3.5.5. Support Vector Machine

Support vector machine (SVM) treats the objects (Harvey tweets) as points in a high-dimensional space and finds a hyperplane to separate them into two categories. Although there are other classifiers based on hyperplanes, SVM has its way of selecting the optimal separation hyperplane [23]. This method (i.e., the maximum-margin hyperplane) maximizes the distance from the separating hyperplane to the nearest data point and finds the largest separation (margin). Not all data are linearly separable, meaning that people cannot always perfectly separate data points, and SVM should be allowed to make classification errors. Considering this, a soft margin formulation is introduced to allow for hyperplane violations while trading off the margin. When introducing a soft margin still cannot separate the data points, the kernel function can be applied to add a new dimension to the dimensional space of the data points, and then a separating hyperplane can be found in this higher-dimensional space. Important hyperparameters of SVM include regularization, kernel, and others. Table A4 lists the hyperparameter settings of this model.

### 3.5.6. Long Short-Term Memory

LSTM (Long short-term memory) is a unique type of RNN (Recurrent Neural Network) that enables the storage of long-term memory via a memory cell unit that has a connection to itself to accumulate external signals [24]. A common LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. This study utilized GloVe embedding based on Twitter from Stanford (open sourced at: <https://nlp.stanford.edu/projects/glove/>, accessed on 20 October 2021). Following the embedding layer, a 1-D convolution layer (128 filters) was first applied in the LSTM model to lower the number of features. A bidirectional layer (128 filters) was then implemented with a dropout ratio of 0.5. Two dense

layers were added following the bidirectional layers, with each of them having 512 filters with the ReLU [25] as the activation function. Finally, a dense layer with one neuron was added, and a sigmoid function was applied, given the binary nature of this classification. Hyperparameter settings for this model can be found in Table A5.

### 3.5.7. Word Embedded Convolutional Neural Network (CNN)

Another competing classification approach in this study is word embedded CNN, originally designed by Kim [26]. In this study, we modified its structure following the architecture design from Huang et al. [18]. Specifically, we obtained word vectors via Word2Vec, a shallow neural network with a single hidden layer but proved to be powerful in providing vectors representing the word characteristics [27]. Please refer to Huang et al. [18] for the detailed word embedded CNN architecture design. Its hyperparameter settings can be found in Table A6.

### 3.6. Model Evaluation

To select the best prediction model, we need to evaluate the performance of the trained machine learning classifiers with test data. Metrics including recall, precision, accuracy, and F1 have been largely used in performance evaluation. These metrics are derived from a two-by-two contingency table (Figure 4) where cells contain the counts of true positive (TP), false positive (FP), true negative (TN), and false negative (FN), respectively.

		Predicted	
		Positive	Negative
Actual	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

**Figure 4.** Contingency table for binary decisions.

The metrics are defined with the following equations:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{TN} + \text{FN})$$

$$\text{F1} = 2 \times \text{Recall} \times \text{Precision} / (\text{Recall} + \text{Precision})$$

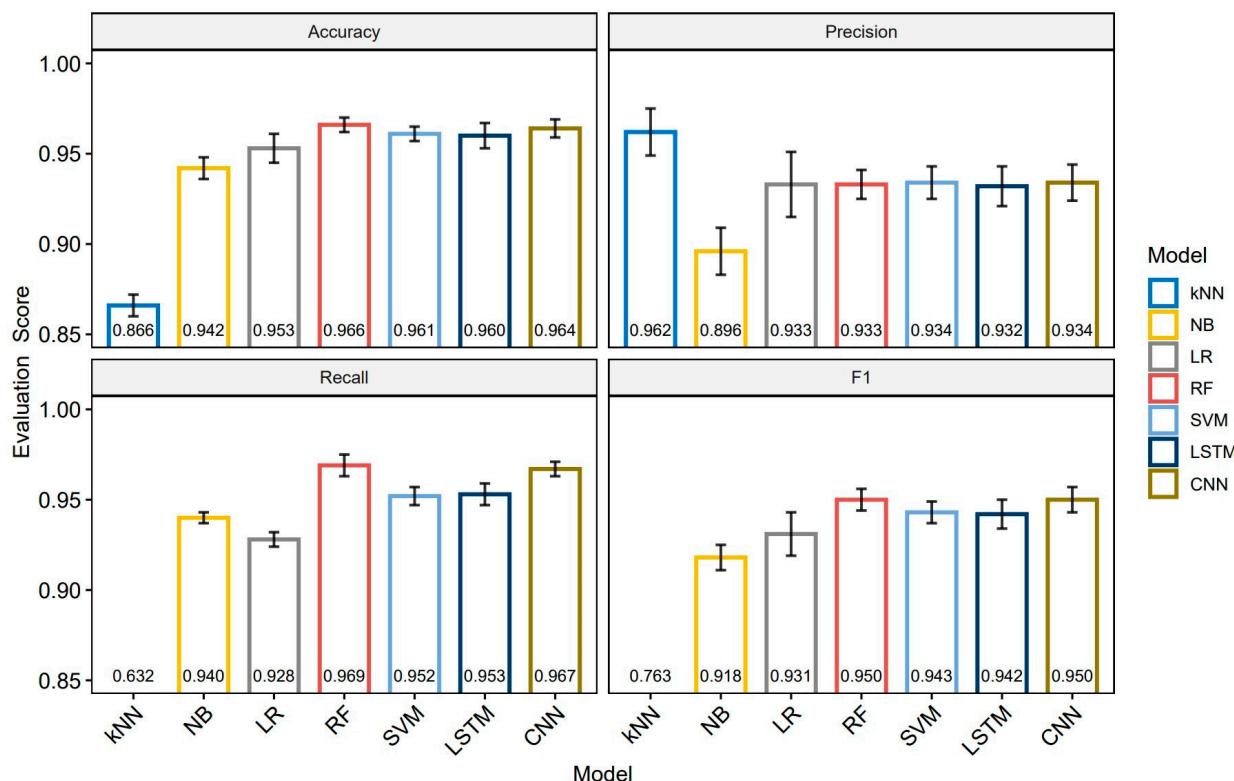
Typically, a higher score indicates that the model performs better. Arguably, the *Recall* score should receive more attention in rescue request detection, because false negatives are preferred to be as low as possible so that not many rescue requests would be missed.

## 4. Results

Note that all positive samples (i.e., *positive\_ZIP*) in the training and test data contain ZIP codes. However, this does not accurately reflect the reality because adding ZIP codes to addresses could be a random act. People would expect that some addresses do not

come with ZIP codes. In light of this, we conducted eight experiments to test how ZIP codes would affect the model performance, where each experiment was based on a dataset created by removing ZIP codes from a certain percentage of *positive\_ZIP*. Notably, the proportion of positive cases that are ZIP-code-untagged is the only difference among the eight datasets, which makes the outcomes of the corresponding experiments comparable. We separated the experiments into two groups: ZIP code majority group and ZIP code minority group. The ZIP code majority group has more than 50% of its positive samples being ZIP-code-tagged, while the minority group has less than 50% of positive samples with ZIP codes.

The training–test split rate for all experiments in this research was set to 80%/20%. To deal with the potential overfitting issue,  $k$ -fold cross-validation was used to split our training data into five ( $k = 5$ ) folds with one fold being a validation set and the remaining four folds as the training set. This split was repeated five times ( $k = 5$ ) so that every fold could serve as the validation set. For each iteration, the model was trained with the training set and evaluated on the validation set. GridSearchCV in scikit-learn was used for cross-validation and hyperparameter tuning. The results of the cross-validation process could be summarized with the mean and standard deviation of the model evaluation scores (see Figure 5 as an example). Below we report the model evaluations obtained with 100% positive cases being ZIP-code-tagged, as well as evaluations of the models with only a portion of positive cases.

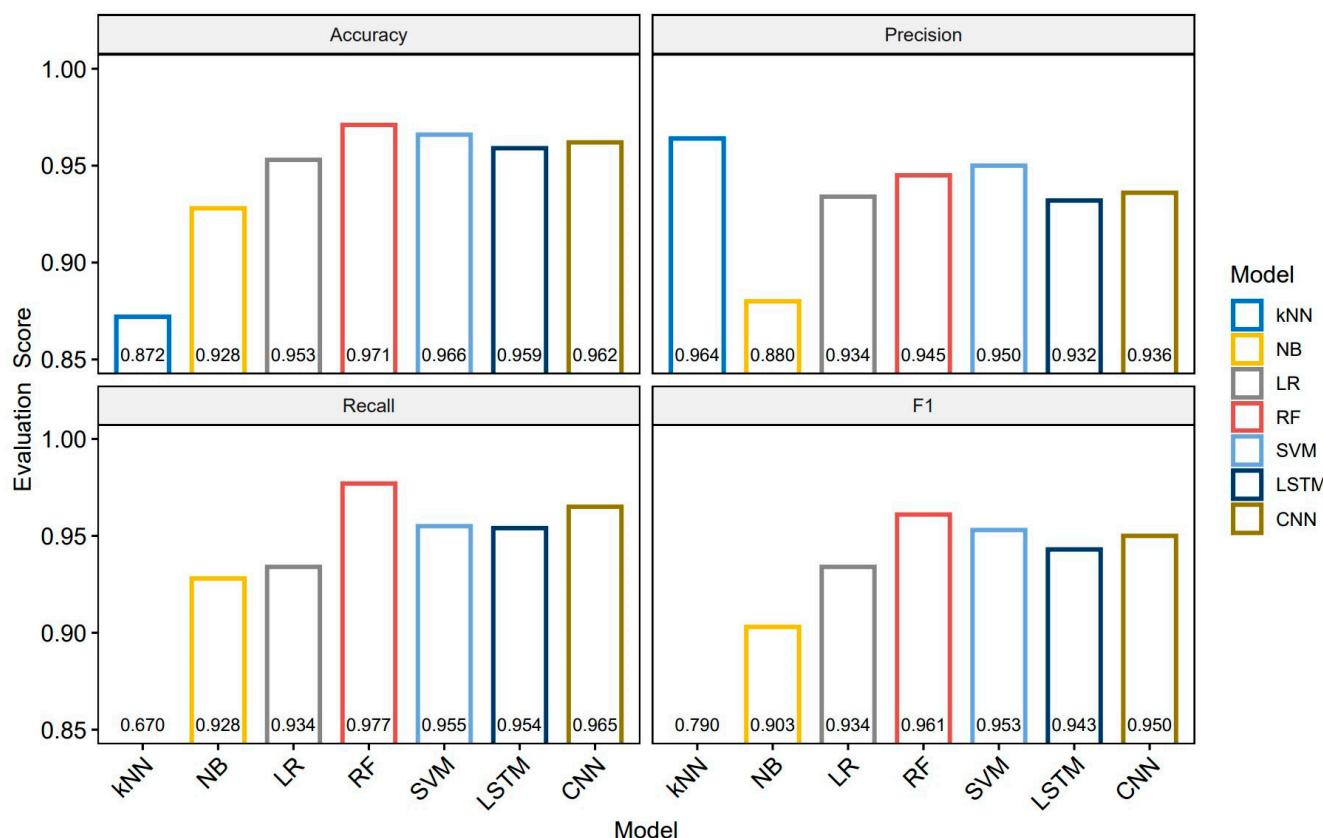


**Figure 5.** Model evaluation results after hyperparameter tuning and cross-validation (bars indicate standard deviations). Note: Although RF and CNN have the same F1 score (0.950), RF has a lower standard deviation (0.006 vs. 0.007).

#### 4.1. Model Evaluation with All Positive Cases Being ZIP-Code-Tagged

As shown in Figure 5, kNN has a significantly higher precision (0.962) at the cost of a low recall rate (0.632). This indicates that it prioritizes classifying Harvey tweets as rescue requests at the cost of a higher false-negative rate. Notably, random forest achieves the highest scores in *Accuracy*, *Recall*, and *F1*; therefore, it is the best classifier in this scenario.

The classifiers obtained with the training data were then further evaluated on the test data. The evaluation results are shown in Figure 6 for comparison. Compared to other models, the random forest classifier again had the best model performance as it achieved the highest scores in *Accuracy*, *Recall*, and *F1*. In other words, random forest was further proved to yield more accurate detection of rescue requests from Harvey tweets. This well-trained random forest classifier was then used to detect rescue requests from the prediction data. Some of the predicted rescue requests were found to be not tagged by ZIP codes (Table 1). Recall that all rescue requests in our training and test data contain ZIP codes in their addresses, and we were unsure if our trained classifier could successfully detect those untagged ones (if any). Therefore, these predicted rescue requests in Table 1, to some extent, validate the effectiveness of our classifier.



**Figure 6.** Model evaluation results after applying trained classifiers to test data.

**Table 1.** Examples of predicted rescue requests.

ID	Predicted Rescue Requests		
1	@TransRightsMOVE @USCG S.O.S. Mentally disabled senior; wife stuck 2nd floor: [REDACTED] Houston		
2	#harveysos #HarveyRescue [REDACTED] @abc13houston @cohoustonfire @houstonpolice		No where to go Water Rising 2 kids 2 adults
3	Family needs rescue (5 and 2 year old) Address: [REDACTED] @abc13houston @Fox26Houston		@houstonpolice @KHOU
4	Y'all my best friend is floating on an air Mattress w her 1 year old baby boy someone please send help [REDACTED]		
5	#cajunNavy #please help #NeedWaterRescue [REDACTED]		-2 infants 2 toddlers 1 child 4 adults 1 dog

Note: We masked the addresses with privacy considerations.

#### 4.2. Model Evaluation with Some Positive Cases Being ZIP-Code-Tagged

We further report the evaluation results after accounting for different ZIP code scenarios in Table 2, where the best model (based on *F1* and *Recall* scores) in each majority group is highlighted in bold and italic font, while the best model in each minority group is highlighted just in bold.

**Table 2.** Model evaluation results with different percentages of ZIP-code-tagged positive samples.

ZIP Code Majority Group: 90% of Requests with ZIP Codes				ZIP Code Minority Group: 90% of Requests without ZIP Codes				
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
kNN	0.961 (0.013)	0.614 (0.022)	0.86 (0.009)	0.749 (0.019)	0.935 (0.003)	0.627 (0.018)	0.858 (0.005)	0.750 (0.012)
NB	0.893 (0.015)	0.940 (0.089)	0.941 (0.007)	0.916 (0.0096)	0.906 (0.010)	0.929 (0.010)	0.943 (0.003)	0.918 (0.005)
LR	0.928 (0.020)	0.927 (0.015)	0.950 (0.006)	0.927 (0.008)	0.937 (0.019)	0.916 (0.009)	0.950 (0.005)	0.927 (0.007)
RF	0.935 (0.004)	0.958 (0.004)	0.963 (0.001)	0.946 (0.001)	0.946 (0.009)	0.939 (0.011)	0.961 (0.005)	0.942 (0.007)
SVM	0.939 (0.004)	0.958 (0.003)	0.944 (0.009)	0.935 (0.009)	0.956 (0.012)	0.929 (0.004)	0.961 (0.005)	0.942 (0.009)
LSTM	0.933 (0.005)	0.957 (0.004)	0.961 (0.002)	0.945 (0.005)	0.947 (0.010)	0.936 (0.005)	0.96 (0.004)	0.941 (0.009)
CNN	<b>0.941 (0.004)</b>	<b>0.959 (0.003)</b>	<b>0.965 (0.001)</b>	<b>0.950 (0.004)</b>	<b>0.953 (0.011)</b>	<b>0.941 (0.007)</b>	<b>0.962 (0.005)</b>	<b>0.947 (0.008)</b>
ZIP Code Majority Group: 80% of Requests with ZIP Codes				ZIP Code Minority Group: 80% of Requests without ZIP Codes				
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
kNN	0.932 (0.009)	0.622 (0.018)	0.856 (0.006)	0.746 (0.013)	0.94 (0.009)	0.618 (0.015)	0.856 (0.005)	0.745 (0.011)
NB	0.897 (0.015)	0.934 (0.011)	0.941 (0.008)	0.915 (0.011)	0.906 (0.010)	0.929 (0.010)	0.943 (0.003)	0.918 (0.005)
LR	0.924 (0.021)	0.916 (0.016)	0.946 (0.012)	0.920 (0.017)	0.937 (0.019)	0.916 (0.009)	0.950 (0.005)	0.927 (0.007)
RF	<b>0.936 (0.007)</b>	<b>0.947 (0.019)</b>	<b>0.960 (0.008)</b>	<b>0.941 (0.012)</b>	0.935 (0.013)	0.927 (0.015)	0.953 (0.005)	0.931 (0.007)
SVM	0.934 (0.015)	0.931 (0.016)	0.954 (0.01)	0.932 (0.015)	0.943 (0.015)	0.925 (0.009)	0.955 (0.003)	0.934 (0.004)
LSTM	0.932 (0.009)	0.945 (0.013)	0.958 (0.009)	0.938 (0.011)	0.941 (0.012)	0.921 (0.011)	0.951 (0.004)	0.931 (0.005)
CNN	0.933 (0.006)	0.944 (0.017)	0.958 (0.008)	0.938 (0.012)	<b>0.945 (0.012)</b>	<b>0.927 (0.007)</b>	<b>0.956 (0.005)</b>	<b>0.936 (0.006)</b>
ZIP Code Majority Group: 70% of Requests with ZIP Codes				ZIP Code Minority Group: 70% of Requests without ZIP Codes				
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
kNN	0.943 (0.016)	0.618 (0.018)	0.857 (0.006)	0.747 (0.013)	0.934 (0.011)	0.615 (0.024)	0.854 (0.008)	0.742 (0.018)
NB	0.906 (0.009)	0.937 (0.012)	0.946 (0.005)	0.921 (0.008)	0.905 (0.012)	0.927 (0.018)	0.942 (0.005)	0.916 (0.007)
LR	0.929 (0.015)	0.925 (0.016)	0.950 (0.005)	0.927 (0.007)	0.931 (0.012)	0.916 (0.017)	0.948 (0.008)	0.924 (0.012)
RF	0.937 (0.008)	0.941 (0.013)	0.958 (0.005)	0.939 (0.008)	<b>0.936 (0.012)</b>	<b>0.941 (0.012)</b>	<b>0.958 (0.007)</b>	<b>0.939 (0.011)</b>
SVM	0.937 (0.010)	0.933 (0.014)	0.956 (0.005)	0.935 (0.007)	0.943 (0.010)	0.927 (0.017)	0.956 (0.007)	0.935 (0.011)
LSTM	0.936 (0.008)	0.941 (0.015)	0.958 (0.006)	0.938 (0.009)	0.935 (0.011)	0.937 (0.013)	0.956 (0.009)	0.936 (0.011)
CNN	<b>0.940 (0.009)</b>	<b>0.942 (0.017)</b>	<b>0.960 (0.005)</b>	<b>0.941 (0.009)</b>	<b>0.940 (0.010)</b>	<b>0.939 (0.014)</b>	<b>0.960 (0.007)</b>	<b>0.940 (0.010)</b>
ZIP Code Majority Group: 60% of Requests with ZIP Codes				ZIP Code Minority Group: 60% of Requests without ZIP Codes				
	Precision	Recall	Accuracy	F1	Precision	Recall	Accuracy	F1
kNN	0.935 (0.015)	0.638 (0.025)	0.862 (0.011)	0.759 (0.022)	0.933 (0.017)	0.613 (0.011)	0.853 (0.005)	0.740 (0.009)
NB	0.900 (0.012)	0.930 (0.01)	0.941 (0.004)	0.915 (0.005)	0.903 (0.012)	0.928 (0.008)	0.941 (0.006)	0.915 (0.008)
LR	0.927 (0.014)	0.912 (0.015)	0.945 (0.005)	0.919 (0.007)	0.929 (0.017)	0.912 (0.021)	0.946 (0.0096)	0.920 (0.015)
RF	0.935 (0.005)	0.947 (0.007)	0.960 (0.002)	0.941 (0.004)	<b>0.934 (0.009)</b>	<b>0.931 (0.017)</b>	<b>0.955 (0.005)</b>	<b>0.937 (0.005)</b>
SVM	0.932 (0.009)	0.928 (0.014)	0.952 (0.006)	0.930 (0.009)	0.939 (0.009)	0.919 (0.014)	0.952 (0.005)	0.929 (0.008)
LSTM	0.936 (0.004)	0.942 (0.009)	0.957 (0.003)	0.939 (0.005)	0.937 (0.010)	0.928 (0.014)	0.953 (0.006)	0.932 (0.010)
CNN	<b>0.932 (0.005)</b>	<b>0.951 (0.010)</b>	<b>0.961 (0.003)</b>	<b>0.941 (0.004)</b>	<b>0.938 (0.011)</b>	<b>0.932 (0.016)</b>	<b>0.955 (0.006)</b>	<b>0.934 (0.009)</b>

Table 2 shows that the best models are all from the ZIP code majority groups if we compare every majority group with its corresponding minority group. The best models in these two groups had *F1* score differ by 0.001 to 0.004. There was a 0.001 to 0.02 difference in the *Recall* score between the top models. Using *t* statistics, we further examined the statistical relevance of the aforementioned differences. Results show that the best models in the majority group had significantly different (statistic = 2.902, *p*-value = 0.027) *Recall* scores from those in the minority group. Although the difference in *F1* was not statistically significant, a marginal improvement in such a life-or-death situation could still save more lives. These experiments could also be taken as sensitivity analysis, which proves that our machine learning models (except kNN and NB) still have excellent performance (above 0.9 for all evaluation scores) after simulating different ZIP code scenarios.

Ranking all the best models based on the *Recall* scores reveals that the top four models are all obtained when ZIP-code-tagged samples account for more than 50% of the entire positive data. Additionally, the positive samples with 100% ZIP-code-tagged train the best detector, while the second-best classifier is obtained with 90% of positive cases coming with ZIP codes. This indicates that ZIP codes are a useful feature for models to learn rescue requests. Although the results do not suggest a perfect linear relationship between ZIP-code-tagged positive cases and the performance of machine learning models, we recommend that every victim tag their rescue requests with ZIP codes due to the following two reasons:

- (1) ZIP-code-tagged rescue requests are more likely to be detected by machine learning models. Therefore, encouraging every rescue-seeking victim to include ZIP codes in their posts would increase the ZIP-code-tagged rescue requests to 90% and above, thus boosting the likelihood of victims' requests being detected.
- (2) Including ZIP codes to make full addresses would enable first responders and volunteers to better locate victims, especially considering that some people may forget to include other geographic information such as cities in their addresses.

## 5. Conclusions

Social media provides a new approach for victims to seek rescue in disasters. Collecting these rescue requests can aid in disaster response. However, most existing classifiers of disaster-related social media data are too general to extract information as specific as rescue requests. This article demonstrates a feasibility study of training a rescue request detector with machine learning and Hurricane Harvey tweets. Accounting for different percentages of ZIP-code-tagged rescue requests in the training process, all machine learning classifiers except kNN and NB achieved excellent performance in differentiating rescue requests from other messages—their model evaluation scores were all above 0.9, offering state-of-the-art performance in the literature where most classifiers' *F1* and *Recall* scores were below 0.9 (see Devaraj, Murthy, and Dontula [11] for the most recent example). This can be explained by the following aspects.

First, our data were purchased from Gnip, which is not subject to the request limit imposed on the free Twitter Application Programming Interfaces (APIs). This enlarged the size of our training data, leading to better learning models.

Second, we focused on actionable rescue requests (must contain geographic addresses), whose features such as 'help', 'rescue', 'rd (road)', and 'dr (drive)' could distinguish themselves from other disaster tweets.

Third and more importantly, we introduced a unique ZIP code filtering method, which significantly reduced the difficulty of screening uncommon rescue requests from a large volume of disaster messages. Compared with existing studies, this research managed to obtain more rescue-seeking tweets even with a stricter definition of rescue requests (must contain geographic addresses) and a smaller study area (Houston MSA). Additionally, after simulating scenarios with different percentages of ZIP-code-tagged positive cases, we found that the best rescue request detector was the one trained with 100% positive samples tagged by ZIP codes, followed by the second-best one trained with 90% ZIP-code-tagged positive samples. Moreover, all the best models in the majority groups outperformed all the best ones in the minority groups. Therefore, overlooking the fact that ZIP codes are an important feature to distinguish rescue requests from other tweets may hinder people from more efficiently retrieving training data and obtaining a better machine learning model to detect rescue requests.

To further illustrate the capacity of the ZIP code filtering, we used the ZIP codes of Louisiana to directly filter rescue requests in Hurricane Ida from the general tweets posted on 30 August 2021. This filtering resulted in 414 tweets. Due to the limitation of resources, these tweets were not manually classified, but a conservative estimation after an arbitrary screening is that more than 40% could be rescue requests. Therefore, we argue

that this straightforward ZIP code filtering can help first responders better retrieve rescue or emergency requests in situations where machine learning classifiers are not available.

Notably, random forest as a traditional machine learning model outperformed two deep learning models in several scenarios, demonstrating that the latter should not always be taken as the best solution without comparing it to traditional methods. This finding is consistent with some existing work [11,28]. For example, Devaraj, Murthy, and Dontula [11] also found that the SVM classifier could have higher  $F1$  than the CNN model in detecting urgent requests in Harvey tweets.

## 6. Discussion

During Hurricane Harvey, rescue requests from social media were manually collected by volunteers and first responders from social media, which is time-consuming and requires intensive human labor. Beyond this feasibility study, our ultimate objective is to develop a generally applicable classifier to automate the detection of rescue requests in various disasters. Future work will test the classifiers with social media messages related to other hurricanes and disasters. For example, we will examine how our classifiers would perform in extracting rescue requests in Hurricane Ida. An interesting research question could be raised accordingly: would the classifiers trained in this study generalize well in another hurricane event?

Many social media analytics including network [29,30], spatial and/or temporal [31–33], and semantic analytics [31–33] have been developed to study Hurricane Harvey. These analytics were often based on general disaster messages, while much less attention has been given to social media activities related to a specific topic (especially rescue requests). Aided by our extraction of rescue requests, these analytics could be enhanced to better understand the social media use in a life-or-death situation and offer more insights into disaster response and resilience.

The approach of considering and testing some features in machine learning modeling may also be relevant to other fields, as social media data have been largely used in many social sciences studies. We advise that, for extraction of a specific type of social media messages during an event such as rescue request messages, it is preferable to identify keywords that are common in target messages but are less common in others. It is also important to conduct simulations to test how models are sensitive to these keywords or features. ZIP codes as a geographic feature have proven useful in improving the machine learning performance in this study. Therefore, geographic features should not be overlooked in machine-learning-based extraction of target messages.

**Author Contributions:** Conceptualization, Zheye Wang; Methodology and software, Zheye Wang, Xiao Huang, and Jin Shang; Formal analysis, Zheye Wang, Xiao Huang, Jin Shang, Yue Wu and Volodymyr V. Mihunov; Data curation, Nina S. N. Lam and Lei Zou; Writing—original draft, Zheye Wang; Writing—review & editing, Zheye Wang, Nina S. N. Lam and Mingxuan Sun; Visualization, Zheye Wang; Supervision, Nina S. N. Lam; Funding acquisition, Nina S. N. Lam and Mingxuan Sun. All authors have read and agreed to the published version of the manuscript.

**Funding:** This article is based on work supported by grants from the National Science Foundation of the United States (under Grant Numbers 1927513, 1620451, 1762600, and 1945787).

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Optimal Hyperparameters

The following tables list the optimal hyperparameters for each model in each experiment. Each experiment here represents a machine learning exercise with a certain percentage of its positive samples containing no ZIP codes. For models except LSTM and CNN, their parameter names are consistent with those in the scikit-learn library in Python, while their other parameters not listed here take the default values as per the

scikit-learn library. Notably, Table A1 shows that kNN always selected the closest neighbor in probability inference to prioritize the *Precision* score over *Recall*, while the latter is more important in our research setting.

**Table A1.** Optimal hyperparameters for the kNN.

Experiment	n_Neighbors
0% no ZIP codes	1
10% no ZIP codes	1
20% no ZIP codes	1
30% no ZIP codes	1
40% no ZIP codes	1
60% no ZIP codes	1
70% no ZIP codes	1
80% no ZIP codes	1
90% no ZIP codes	1

**Table A2.** Optimal hyperparameters for the LR.

Experiment	C	Penalty	Solver
0% no ZIP codes	10	l2	liblinear
10% no ZIP codes	10	l2	liblinear
20% no ZIP codes	10	l2	liblinear
30% no ZIP codes	10	l2	liblinear
40% no ZIP codes	10	l2	liblinear
60% no ZIP codes	10	l2	liblinear
70% no ZIP codes	10	l2	liblinear
80% no ZIP codes	10	l2	liblinear
90% no ZIP codes	10	l2	liblinear

**Table A3.** Optimal hyperparameters for the RF.

Experiment	n_Estimators	Max_Features	Max_Depth	Min_Samples_Split	Min_Samples_Leaf
0% no ZIP codes	200	log2	300	20	1
10% no ZIP codes	400	log2	300	10	1
20% no ZIP codes	600	log2	200	10	1
30% no ZIP codes	300	log2	100	15	1
40% no ZIP codes	100	log2	300	12	1
60% no ZIP codes	200	log2	300	10	1
70% no ZIP codes	400	log2	300	12	1
80% no ZIP codes	400	log2	200	15	1
90% no ZIP codes	500	log2	150	10	1

**Table A4.** Optimal hyperparameters for the SVM.

Experiment	C	Kernel	Gamma
0% no ZIP codes	1	rbf	1
10% no ZIP codes	1	rbf	1
20% no ZIP codes	1	rbf	1
30% no ZIP codes	1	rbf	1
40% no ZIP codes	5	rbf	1
60% no ZIP codes	1	rbf	1
70% no ZIP codes	5	rbf	1
80% no ZIP codes	5	rbf	1
90% no ZIP codes	1	rbf	1

**Table A5.** Hyperparameters settings for the LSTM.

Experiment	Embedding Dimension	Batch Size	Learning Rate	Epoch Cap	Early Stop
0% no ZIP codes	300	64	0.0001	100	Yes
10% no ZIP codes	300	64	0.0001	100	Yes
20% no ZIP codes	300	64	0.0001	100	Yes
30% no ZIP codes	300	64	0.0001	100	Yes
40% no ZIP codes	300	64	0.0001	100	Yes
50% no ZIP codes	300	64	0.0001	100	Yes
60% no ZIP codes	300	64	0.0001	100	Yes
70% no ZIP codes	300	64	0.0001	100	Yes
80% no ZIP codes	300	64	0.0001	100	Yes
90% no ZIP codes	300	64	0.0001	100	Yes

**Table A6.** Hyperparameter settings for the CNN.

Experiment	Dimensions of Input Vectors	Concatenated Vector Length	Batch Size	Learning Rate	Epoch Cap	Early Stop
0% no ZIP codes	300	1024	64	0.001	100	Yes
10% no ZIP codes	300	1024	64	0.001	100	Yes
20% no ZIP codes	300	1024	64	0.001	100	Yes
30% no ZIP codes	300	1024	64	0.001	100	Yes
40% no ZIP codes	300	1024	64	0.001	100	Yes
50% no ZIP codes	300	1024	64	0.001	100	Yes
60% no ZIP codes	300	1024	64	0.001	100	Yes
70% no ZIP codes	300	1024	64	0.001	100	Yes
80% no ZIP codes	300	1024	64	0.001	100	Yes
90% no ZIP codes	300	1024	64	0.001	100	Yes

## References

1. Mihunov, V.V.; Lam, N.S.N.; Zou, L.; Wang, Z.; Wang, K. Use of Twitter in Disaster Rescue: Lessons Learned from Hurricane Harvey. *Int. J. Digit. Earth* **2020**, *13*, 1454–1466. [[CrossRef](#)]
2. Zou, L.; Lam, N.S.N.; Shams, S.; Cai, H.; Meyer, M.A.; Yang, S.; Lee, K.; Park, S.J.; Reams, M.A. Social and Geographical Disparities in Twitter Use during Hurricane Harvey. *Int. J. Digit. Earth* **2019**, *12*, 1300–1318. [[CrossRef](#)]
3. Wang, Z.; Ye, X.; Tsou, M.H. Spatial, Temporal, and Content Analysis of Twitter for Wildfire Hazards. *Nat. Hazards* **2016**, *83*, 523–540. [[CrossRef](#)]
4. Wang, Z.; Ye, X. Social Media Analytics for Natural Disaster Management. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 49–72. [[CrossRef](#)]
5. Wang, Z.; Ye, X. Space, Time, and Situational Awareness in Natural Hazards: A Case Study of Hurricane Sandy with Social Media Data. *Cart. Geogr. Inf. Sci.* **2019**, *46*, 334–346. [[CrossRef](#)]
6. Wang, Z.; Lam, N.S.N.; Obradovich, N.; Ye, X. Are Vulnerable Communities Digitally Left behind in Social Responses to Natural Disasters? An Evidence from Hurricane Sandy with Twitter Data. *Appl. Geogr.* **2019**, *108*, 1–8. [[CrossRef](#)]
7. Huang, X.; Wang, C.; Li, Z. Reconstructing Flood Inundation Probability by Enhancing near Real-Time Imagery with Real-Time Gauges and Tweets. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4691–4701. [[CrossRef](#)]
8. Zou, L.; Lam, N.S.N.; Cai, H.; Qiang, Y. Mining Twitter Data for Improved Understanding of Disaster Resilience. *Ann. Am. Assoc. Geogr.* **2018**, *108*, 1422–1441. [[CrossRef](#)]
9. Hughes, A.L.; Palen, L. The Evolving Role of the Public Information Officer: An Examination of Social Media in Emergency Management. *J. Homel. Secur. Emerg. Manag.* **2012**, *9*. [[CrossRef](#)]
10. Imran, M.; Mitra, P.; Castillo, C. Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages. *arXiv* **2016**, arXiv:1605.05894.
11. Devaraj, A.; Murthy, D.; Dontula, A. Machine-Learning Methods for Identifying Social Media-Based Requests for Urgent Help during Hurricanes. *Int. J. Disaster Risk Reduct.* **2020**, *51*, 101757. [[CrossRef](#)]
12. Kabir, M.Y.; Madria, S. A Deep Learning Approach for Tweet Classification and Rescue Scheduling for Effective Disaster Management. In Proceedings of the GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, Chicago, IL, USA, 10 August 2010; Association for Computing Machinery: New York, NY, USA, 2019; pp. 269–278.
13. Imran, M.; Elbassuoni, S.; Castillo, C.; Diaz, F.; Meier, P. Extracting information nuggets from disaster-Related messages in social media. *Iscram* **2013**, *201*, 791–801.

14. Huang, Q.; Xiao, Y. Geographic Situational Awareness: Mining Tweets for Disaster Preparedness, Emergency Response, Impact, and Recovery. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 1549–1568. [\[CrossRef\]](#)
15. Nie, J.-Y. Institute of Electrical and Electronics Engineers. In Proceedings of the IEEE Computer Society 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017; ISBN 9781538627150.
16. Zhou, B.; Zou, L.; Mostafavi, A.; Lin, B.; Yang, M.; Gharaibeh, N.; Cai, H.; Abedin, J.; Mandal, D. VictimFinder: Harvesting Rescue Requests in Disaster Response from Social Media with BERT. *Comput. Env. Urban Syst.* **2022**, *95*, 101824. [\[CrossRef\]](#)
17. de Albuquerque, J.P.; Herfort, B.; Brenning, A.; Zipf, A. A Geographic Approach for Combining Social Media and Authoritative Data towards Identifying Useful Information for Disaster Management. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 667–689. [\[CrossRef\]](#)
18. Huang, X.; Li, Z.; Wang, C.; Ning, H. Identifying Disaster Related Social Media for Rapid Response: A Visual-Textual Fused CNN Architecture. *Int. J. Digit. Earth* **2020**, *13*, 1017–1039. [\[CrossRef\]](#)
19. Huang, X.; Wang, C.; Li, Z.; Ning, H. A Visual-Textual Fused Approach to Automated Tagging of Flood-Related Tweets during a Flood Event. *Int. J. Digit. Earth* **2019**, *12*, 1248–1264. [\[CrossRef\]](#)
20. Abu-Nimeh, S.; Nappa, D.; Wang, X.; Nair, S. A comparison of machine learning techniques for phishing detection. In Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, Pittsburgh, PA, USA, 4–5 October 2007; pp. 60–69.
21. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos Pedregosa, Varoquaux, Gramfort et al. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
22. Soria, D.; Garibaldi, J.M.; Ambrogi, F.; Biganzoli, E.M.; Ellis, I.O. A “non-Parametric” Version of the Naive Bayes Classifier. *Knowl. Based Syst.* **2011**, *24*, 775–784. [\[CrossRef\]](#)
23. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [\[CrossRef\]](#)
24. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*. 2012. Available online: [https://www.isca-speech.org/archive\\_v0/archive\\_papers/interspeech\\_2012/i12\\_0194.pdf](https://www.isca-speech.org/archive_v0/archive_papers/interspeech_2012/i12_0194.pdf) (accessed on 18 February 2022).
25. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
26. Yoon, K. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; pp. 1746–1751.
27. Tomas, M.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
28. Jiao, S.; Gao, Y.; Feng, J.; Lei, T.; Yuan, X. Does Deep Learning Always Outperform Simple Linear Regression in Optical Imaging? *Opt. Express* **2020**, *28*, 3717. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Rajput, A.A.; Li, Q.; Zhang, C.; Mostafavi, A. Temporal Network Analysis of Inter-Organizational Communications on Social Media during Disasters: A Study of Hurricane Harvey in Houston. *Int. J. Disaster Risk Reduct.* **2020**, *46*, 101622. [\[CrossRef\]](#)
30. Fan, C.; Jiang, Y.; Yang, Y.; Zhang, C.; Mostafavi, A. Crowd or Hubs: Information Diffusion Patterns in Online Social Networks in Disasters. *Int. J. Disaster Risk Reduct.* **2020**, *46*, 101498. [\[CrossRef\]](#)
31. Yang, J.; Yu, M.; Qin, H.; Lu, M.; Yang, C. A Twitter Data Credibility Framework—Hurricane Harvey as a Use Case. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 111. [\[CrossRef\]](#)
32. Havas, C.; Resch, B. Portability of Semantic and Spatial-Temporal Machine Learning Methods to Analyse Social Media for near-Real-Time Disaster Monitoring. *Nat. Hazards* **2021**, *108*, 2939–2969. [\[CrossRef\]](#)
33. Chen, S.; Mao, J.; Li, G.; Ma, C.; Cao, Y. Uncovering Sentiment and Retweet Patterns of Disaster-Related Tweets from a Spatiotemporal Perspective—A Case Study of Hurricane Harvey. *Telemat. Inform.* **2020**, *47*, 101326. [\[CrossRef\]](#)