Summer 2022

Social media analytics with applications in disaster management and COVID-19 events

Md Yasin Kabir

Department: Computer Science

SOCIAL MEDIA ANALYTICS WITH APPLICATIONS IN DISASTER MANAGEMENT AND COVID-19 EVENTS

by

MD YASIN KABIR

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2022

Approved by:

Sanjay Madria, Advisor
Tony Luo
Ardhendu Tripathy
A. Ricardo Morales
Cihan H. Dagli

PUBLICATION DISSERTATION OPTION

This dissertation consists of the following four articles which have been either published or submitted for publication, formatted in the style used by the Missouri University of Science and Technology.

Paper I: found on pages 21–52, Kabir, Md Yasin, and Sanjay Madria. "A deep learning approach for tweet classification and rescue scheduling for effective disaster management", was published in Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 2019.

Paper II: found on pages 53–74, Kabir, M. Yasin, Sergey Gruzdev, and Sanjay Madria. "STIMULATE: A System for Real-time Information Acquisition and Learning for Disaster Management", was published in 2020 21st IEEE International Conference on Mobile Data Management (MDM). IEEE, 2020.

Paper III: found on pages 75–110, Kabir, Md Yasin, and Sanjay Madria. "EMOCOV: Machine learning for emotion detection, analysis and visualization using COVID-19 tweets", was published in Online Social Networks and Media 23 (2021).

Paper IV: found on pages 111–133, Kabir, Md Yasin, and Sanjay Madria. "A deep learning approach for ideology detection and polarization analysis during the COVID-19 pandemic leveraging social media." has been submitted to 41st International Conference on Conceptual Modeling (ER 2022).

ABSTRACT

Social media such as Twitter offers a tremendous amount of data throughout an event or a disastrous situation. Leveraging social media data during a disaster is beneficial for effective and efficient disaster management. Information extraction, trend identification, and determining public reactions might help in the future disaster or even avert such an event. However, during a disaster situation, a robust system is required that can be deployed faster and process relevant information with satisfactory performance in real-time. This work outlines the research contributions toward developing such an effective system for disaster management, where it is paramount to develop automated machine-enabled methods that can provide appropriate tags or labels for further analysis for timely situation-awareness. In that direction, this work proposes machine learning models to identify the people who are seeking assistance using social media during a disaster and further demonstrates a prototype application that can collect and process Twitter data in real-time, identify the stranded people, and create rescue scheduling. In addition, to understand the people's reactions to different trending topics, this work proposes a unique auxiliary feature-based deep learning model with adversarial sample generation for emotion detection using tweets related to COVID-19. This work also presents a custom Q&A-based RoBERTa model for extracting related phrases for emotions. Finally, with the aim of polarization detection, this research work proposes a deep learning pipeline for political ideology detection leveraging the tweet texts and the expressed emotions in the text. This work also studies and conducts the historical emotion and polarization analysis of the COVID-19 pandemic in the USA and several individual states using tweeter data.

ACKNOWLEDGMENTS

All praise to the almighty for giving me the ability and privilege to finish my Ph.D. It was a long journey on which I received exceptional support from my family, friends, and colleagues.

I am highly thankful to my supervisor Dr. Sanjay Madria. His continuous support and guidance were remarkable. His unwavering encouragement made it possible to finish my dream while having two kids. I would like to thank all of my Ph.D. committee members: Dr. Dagli, Dr. Luo, Dr. Tripathy, and Dr. Morales for taking time and interest in my work and for their insightful suggestions. I am thankful to Dr. Bojan Tunguz and Dr. Ramon Michael Alvarez for their mentorship and guidance during my internship. I want to take a moment to thank all of my teachers who guided me from my early childhood.

I am grateful to my labmates Azhar, Shudip, Ayan, Xiofei, and others for their help and feedback during my Ph.D. I am very grateful to Dr. Shaikat Galib who inspired me a lot in machine learning and helped me to grasp the core concepts. I am thankful to NSTU CSTE 4th batch and the Galactica team who helped and motivated me to pursue this journey in so many ways. I learned a lot from two of my seniors Towhid and Bulbul which inspired me to dive deep and pursue my interest in higher education.

I express my heartfelt indebtedness to my family members for their sacrifices. The unconditional love and support from my parents always kept me motivated during this challenging journey. My parents always prioritized my needs before their happiness and ensured the necessary environment and best possible education for me.

I am extremely thankful to my life partner Ajibun Nahar Eva. My wife sacrificed her education and career to stay with me. It wouldn’t be possible for me to finish this journey without her unconditional support. She earned this achievement with me. I dedicate this dissertation to her.

TABLE OF CONTENTS

Page

SECTION

PAPER

SECTION

LIST OF ILLUSTRATIONS

Figure Page

SECTION

PAPER I

PAPER II

PAPER III

PAPER IV

LIST OF TABLES

Table Page

PAPER I

PAPER II

PAPER III

PAPER IV

SECTION

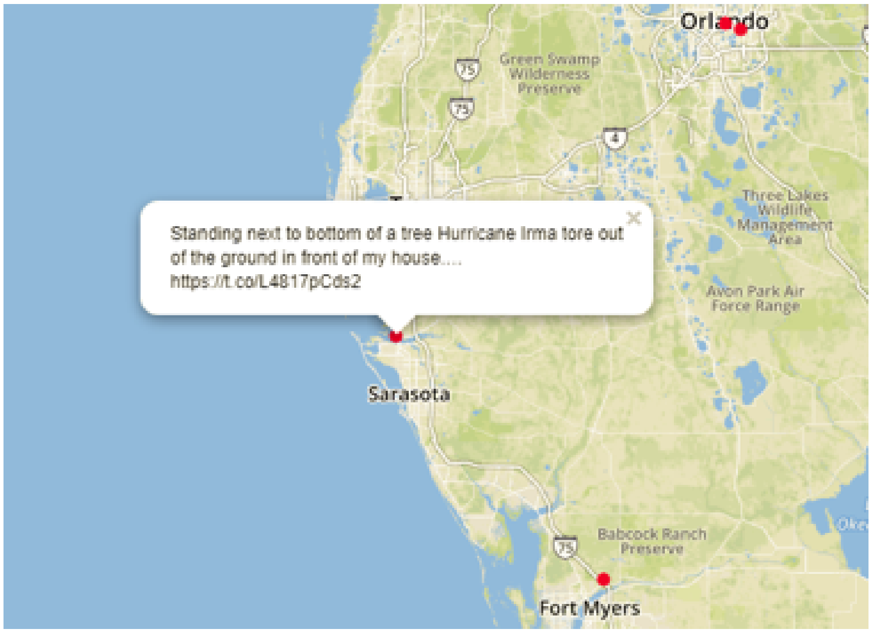
1. INTRODUCTION

A disaster can refer to an effect and result of natural or human made events such as flood, hurricane, earthquake, tornado, heatwave, pandemic, riot, terrorist attack, etc. Disaster management has three critical phases (planning, response, and recovery) and effectiveness of each phase depends on accurate and up-to-date information. “Social Media Data Mining” can be referred as a form of crowdsourcing that take leverage of social media data (e.g. Twitter, Facebook) to detect an event and further understand and validate the evolution and impact of that event. Because of huge adaptation of social media by mass people, researcher found that social media data mining can be beneficial for event detection, response, communication, and monitoring during crisis and emergency situations (e.g. disease outbreaks [68, 23], hurricanes and floods [102, 97, 92, 90], rumor and violence [13, 84, 72]). Apart from active disaster management and monitoring Twitter data also proven very critical in human sentiment analysis which be useful for predict crimes [17, 27], community mental health [80, 91, 70], polarity [34, 73], etc. Although there are many existing research works on social media data mining and analysis, many of those research works need further exploration and improvement. Furthermore, with the growing use and adaptation of social media, new research problems and opportunities are emerging faster. There is a lack of effective machine learning models for social media disaster text classification. Another major challenge is to find or prepare appropriate task-specific labeled data for machine learning. During some crises (e.g. The ongoing COVID-19 pandemic) human emotion plays a crucial role. However, human emotion detection from microblogs (e.g., Tweets) is very challenging due to the nature of complex emotional expressions in a short text. While some existing methods work well for a few events, they failed to adapt and generalize for the new events.

1.1. USE CASES AND CHALLENGES

1.1.1. Disaster Management. Disaster management is a combination of three essential phases (Planning, Response, and Recovery). Communication plays a vital role in all three of those phases. Fast and effective communication can sometimes prevent a disaster or able to reduce the impact. On the contrary, poor communication or lack of communication makes it challenging to manage a disaster and causes an unintended outcome. Natural disasters frequently disrupt regular communication [82] and hindered the essential information flow. Besides, the use of traditional calls or text messages as communication during a disaster might not be optimal. With the growth of the internet, people use non-traditional mediums such as social media for faster communication. To get information quickly (e.g. weather forecast, alerts, shelter info) during a disaster, social media, websites became more preparable [8].

Acquiring real-time updated information is required for effective disaster management. For instance, *Floods* are one of the most common natural disasters that can cause significant damage to life, infrastructures, and the economy. Early predictions and real-time flood routing detection can play a crucial role to save lives and resources [60]. Real-time flood information mapping can assist in safe evacuation and rescue operations. With the advent of technologies and techniques, it is now possible to provide early flood prediction and probable flood route mapping. However, real-time flood mapping and information acquiring still challenging. In an adverse disaster situation, satellite images can be foggy, blurry and might not able to provide useful information of the flood-affected areas.



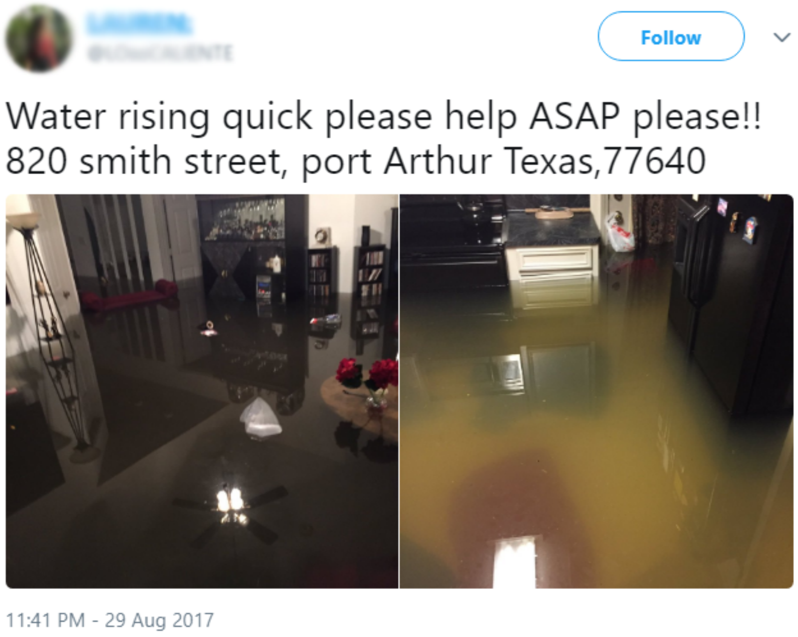
Standing next to bottom of a tree Hurricane Irma tore out of the ground in front of my house.... https://t.co/L4817pCds2

Orlando
Green Swamp Wilderness Preserve
75
75
4
Three Lakes Wildlife Management Area
Avon Park Air Force Range
Sarasota
Babcock Ranch Preserve
75
Fort Myers
Oke

Figure 1.1. Example of informative tweets during a disaster

Social sensing can be useful to fill the gap and gather near real-time information during a flood. [25]. *Social sensing* can be referred to as a form of crowd sourcing that takes leverage of social media data (e.g. Twitter, Facebook) to detect an event and further understand and validate the evolution and impact of that event. Social media data such as tweets can also provide useful information about transport disruption (e.g. damaged roads, blocked roads due to fallen trees or infrastructures), flow and the level of the floodwater, and other hazard events during a disaster. Figure 1.1 represents an example of informative and useful tweet about fallen tree during Hurricane Irma. Apart from that, social media is also proven useful in disaster management for various crises such as infected disease [23] and violence [13, 84].

1.1.2. Twitter Data Mining to Assist People during a Disaster. Natural disasters frequently disrupt the regular communication due to the damaged infrastructures which leads to an outflow of required information. Non-traditional communication systems like social media become more interactive and provide a more continuous flow of information. During Hurricane "Sandy" [8] people used social media more frequently to communicate



[redacted]

Water rising quick please help ASAP please!!
820 smith street, port Arthur Texas,77640

11:41 PM - 29 Aug 2017

Figure 1.2. Rescue requesting tweets example

and seeking help. Social media allows people to communicate promptly and facilitate information regarding transport, shelter, and food. Therefore, the huge flow of information over social media can be beneficial to manage a disaster.

Social media websites such as Twitter and Facebook experiencing mass adaptation and exponential growth. The features and services provided by those websites have a broader impact and is not limited to general social communication. The roles of social media extended but not limited to health and disease analysis and propagation detection [75, 81, 30], Quantifying controversy [26], finance market indicator [48], information and rumor flowing [78, 65], disaster crisis management [4, 97]. During the Hurricane Sandy, Twitter proved it's usefulness and at the time of Hurricane Harvey and Irma Twitter played a crucial role in the rescue, donation, and recovery [54]. Figure 1.2 represent a tweet asking help during Hurricane Harvey while also providing critical information about the flood. While there are enormous use cases of social media for disaster management, however, still there is a lack of disaster management tools that leverage the social media data for effective disaster management. The primary challenge is to process enormous amount of data in real-time and extract the useful information from that data.

1.1.3. Tweets Emotion Classification. Social media data and human emotion analysis can be highly useful in disaster management [97, 8], crime prediction [17, 27], tracking stock market sentiments [69, 35], detecting polarity [34], etc. Therefore, sentiments analysis [7] became a popular field of natural language processing. There is a wide range of research works available where sentiments are explored using different methods and tools. During the COVID-19 pandemic, social distancing or stay-at-home became the most widely used directive all over the world. Social distancing is impacted almost every activity associated with human life. People lost their jobs and earning sources. Thus, the emotional responses became overwhelming due to this unprecedented event. Researchers have observed an increasing rate of toxic comments and political polarization during the pandemic. Hence, the exploration of tweets to track emotions might play a significant role to understand people's behaviors and responses during an event such as the COVID-19 pandemic.

In most of the sentiment analysis work, sentiments are explored considering high-level emotion categories such as positive, neutral, and negative. Several works also considered sentiment as a form of feeling using numerical scores. However, to understand the emotional response accurately we need fine-grained labels of emotion. For example, instead of a generic label such as negative sentiment, labeling the emotions like sadness, worry, or angry might enable us to understand the proper reaction of a person as each of those individual emotions might exhibit different behavior. While identifying fine-grained emotion is intriguing, it is also challenging due to the lack of appropriate data sets. Therefore, most of the research works on COVID-19 sentiment analysis are limited to high-level emotions (positive, neutral, and negative). To address this challenge a dataset was processed and annotated for this research work. The experiments were conducted using the annotated data along with the publicly available external data sets. Another unique dataset also created where a phrase related to identified emotion was selected as the key words for the expressed emotion. This selected phrase played critical role in further understanding and analysis of the emotion.

Another major challenge in tweets emotion classification is to train a model which can understand the context. Because of the unbalanced dataset and ever-evolving nature of human texting, it is difficult to map the attributes in a desirable way that can understand the underneath context. Besides, it is proven difficult to determine the appropriate emotion from a text by an expert human annotator also. To address this problem, we developed an auxiliary feature-based classifier along with the text which can give some numerical attributes to the models. Further, we have also proposed a module to leverage the adversarial sample generation technique and generate samples to train the model. This approach effectively increased the performance of the emotion classification model and improved the accuracy in the minor classes with lower original annotated samples.

1.1.4. Polarization Detection. Understanding polarization is very critical to motivating the mass population effectively and creating acceptable policies. Polarization can act as an obstacle to achieving something for the greater good. Polarization also drives hate speech and sometimes conflicts which can be categorized as a disaster. A proper study of polarization for some topics can be useful for other topics that might arise in the future. It will also assist in sharing acceptable information across all demographic. The use of social media platforms increased extensively during the COVID-19 pandemic because of isolation and stay-at-home directives. Almost all types of social media users including but not limited to personal accounts, business pages, news, institutions, and government officials leverage social media to share information, and directives to generate awareness among the mass population. People shared their opinions, beliefs, and political agendas along with various content types. Various topics related to Covid-19 with great interest have emerged throughout the pandemic. "Mask, Vaccine, Stay at home" are some examples of the topics with high engagement. People with different ideologies reacted differently to those topics. During the unprecedented COVID-19 pandemic, researchers have observed a concerning amount of polarization, racial comments, and hate crimes in social media [47, 45] that can correlate with several disastrous events during this period. During the pandemic researchers have found polarization and conspiracies for general health directives such as masks and vaccines [15, 98].

While there are research works on COVID-19 polarization detection [98, 46, 32], most of those works use relatively small data sets or periods. However, those work shows that polarization is highly visible across different political ideologies. Researchers mostly take two different approaches to political ideology detection. One is content-based and another one is network-based. In both approaches, a set of seed users are determined based on publicly known political affiliation. Politicians, journalists, and verified users with self-claimed political beliefs are considered the seed users. In a content-based approach, user profile information such as location, workplace, gender, age, and shared content is used to determine the political ideology. In the network-based approach interaction between the seed users and other people is considered to form a network. People who are densely connected and interact much with each other are considered to have the same ideology. While both of the methods work well depending on the use cases, they also face challenges due to the nature of the approach. Content-based approach suffers from the location and demographic bias along with the content similarity where contents in two articles might be similar but express different opinions. The network-based approach face challenge when people with opposite views interact with each other. It might be possible for two groups of people to engage in an interaction extensively for a topic of interest. In this research work, a content-based approach is taken to determine the political ideology. However, in our approach, we use only the tweet text instead of considering any user profile or interaction information. We leverage the deep neural network to extract the emotion in the tweet and further, we train a transformer-based network to understand the context of the tweets in order to determine the political ideology expressed in the tweet.

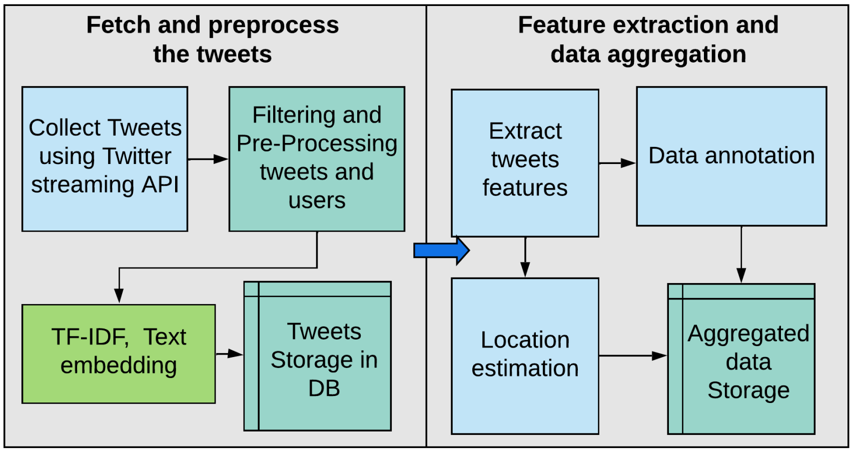

Fetch and preprocess the tweets
Collect Tweets using Twitter streaming API
Filtering and Pre-Processing tweets and users
TF-IDF, Text embedding
Tweets Storage in DB
Feature extraction and data aggregation
Extract tweets features
Data annotation
Location estimation
Aggregated data Storage

Figure 1.3. Data processing overview

1.2. DATA COLLECTION AND PROCESSING

Real-time data collection, processing, and analysis are critical for effective disaster management. Due to extensive use, social media platforms produce an enormous amount of data every day. In this research work, Twitter is used as the primary data source for all of the analyses and experiments. The design and development of an effective and efficient data collection and processing pipeline was one of the primary aims of this research work. To achieve this goal efficient data processing pipelines and applications were built over the CPU and GPU. Figure 1.3 presents the general overview of the data collection, processing, feature extraction, and data annotation pipeline. It primarily consists of 5 modules/Subtasks that are described below.

1. **Tweet Fetcher:** This module uses Twitter Streaming API to obtain a stream of the tweets in real-time on specified topics. This module collects the raw tweets and saves those tweets as plain text. Twitter API provides the tweets in dictionary format which is similar to JSON format. To obtain the specific tweets for a specific user or with the specific ID of the tweets the Twitter search API is also used along with

the Twitter streaming API. Using this module we have collected tweets since 2017 during several disasters notably hurricane Harvey and Hurricane Irma. We have also collected tweets during the covid-19 pandemic from March 2020 to December 2021.

2. **Tweet Pre-processor:** The tweet pre-processor module filters unexpected or invalid tweets based on keywords and language. This module also identified the tweets with media attachments or web URLs and tag those features. Using a web scrapping script this module also collects the media in the tweets and saves those in the storage. The pre-processing module also filters the jargon and removes duplicated entries. Depending on the use case this module calls a sub-module that performs topic modeling and text embedding for the primary analysis.

3. **Feature Extraction:** In this step, the necessary features from the tweets are extracted. This involved a semi-automated feature engineering process to identify the valuable features that provide maximum information for the research task while reducing the data size effectively. A set of auxiliary features from the tweet text is also extracted in this step. Those features included but were not limited to sentiment, number of punctuation, emojis, number of capital letters and words, number of hashtags, unique words, parts of speech tags, etc.

4. **Location estimation:** Location plays a critical role in proper data analysis. It also assists to understand the variability in different geographical regions. Due to Twitter's data privacy policy, only a fraction of tweets contain the location. There we leverage the user profile meta information to extract the high-level location of the respective user. During the disasters, we have observed that people specify the location/address in the text of the tweet while seeking assistance. This module uses Stanford Named Entity Recognizer (NER) [24] to extract the address within the tweet text. Along with the NER, this module also uses Google Maps API to extract and estimate the location of a user from the tweet text.

5. **Data Annotation:** There is a lack of available datasets for tweet classification and emotion detection. To develop effective deep learning models quality data is essential. Data annotation is one of the major contributions of this research work. Manual and semi-automatic data annotation methods are used for data annotation. We have annotated four different datasets for model development, training, and testing. Further, we made those datasets publicly available to the research community. The detailed process of the data annotation is described in the papers included in this dissertation.

1.3. DATA PROCESSING OVER THE GPU

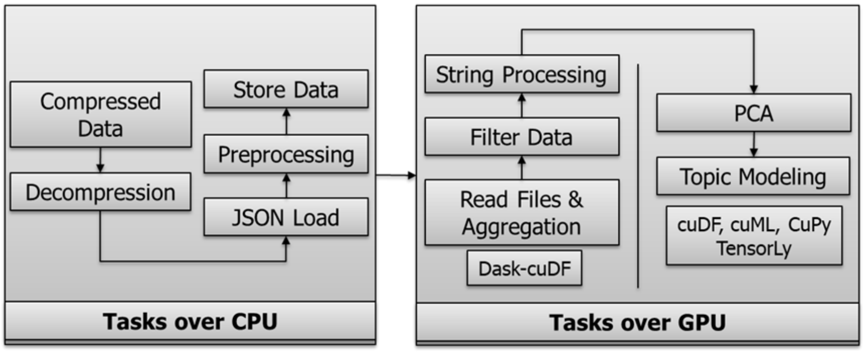


Figure 1.4. GPU processing

Data processing is crucial for any data analytic and machine learning project. This research project involves terabytes of Twitter data and sometimes a tremendous amount of streaming data needed to be processed in real-time. Processing a large dataset in a short term required efficient use of storage, multi-processing, and parallel processing. To be able to process data fast we leveraged distributed computing over cloud infrastructure, multi-processing over CPUs, and parallel use of GPUs. During our research, we found that filtering and processing an immense amount of text data over CPU requires a tremendous amount of time which can be effectively reduced by leveraging consumer-grade GPU. Processing data

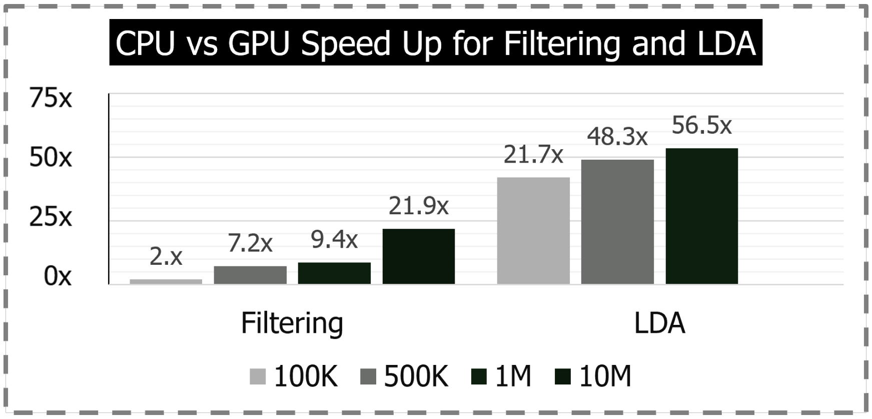


Figure 1.5. Performance Comparison between CPU and GPU

over GPU also reduces the computational cost and it requires fewer resources compared to the CPU cluster. In this research work, we have used existing methods for string processing and topic modeling to adopt those over GPU.

Figure 1.4 represents the overall data processing workflow. In the figure tasks performed over CPU and GPU are separated for the ease of understanding. However, those tasks were performed parallelly in both CPU and GPU. While GPUs can process the data very fast, there is still some limitation. Due to the volatile implementation of GPU the data processed by the GPU doesn't always produce a sequential output. For example, the data is split over the multiple GPU cores and when it combines those data the sequence becomes random. To store those data in the storage with correct order, I had to use some methods that run over CPU. This while acts as a bottleneck, yet the combine processing outperformed the performance of a larger number of CPU cores. Figure 1.4 presents a performance comparison for two tasks between CPU and GPU. For the experiments a machine comprises of Intel® Core™ i9-9900K CPU (8 cores, 16 CPU), 64GB RAM and an Nvidia RTX-2080Ti GPU is used. For filtering the data and topic modeling using LDA the GPU can speed up over 21x (10M data) and 56x (1M) consecutively.

1.4. DISSERTATION SUMMARY

This dissertation is composed of four papers presented in publication format of the conference or the journal wherein they were published or submitted addressing the aforementioned objectives in the previous sections.

Paper I titled "A deep learning approach for tweet classification and rescue scheduling for effective disaster management" presents a multi-headed binary classifier that classifies the tweets into six different classes to detect the stranded people during hurricanes Harvey and Irma. This paper proposes a unique deep learning pipeline utilizing a set of punctuation-based auxiliary features. Further, a multi-task hybrid scheduling algorithm is introduced for rescue scheduling that leverages the output labels from the classifier.

Paper II titled "STIMULATE: A System for Real-time Information Acquisition and Learning for Disaster Management" explores and utilizes the proposed deep learning model and scheduling algorithm in Paper I. This paper describes STIMULATE which can facilitate real-time social media data collection, processing, and rescue management. The system also introduces an easy-to-use web interface where institutions and individuals can participate in rescue efforts with real-time synchronization.

Paper III titled "EMOCOV: Machine learning for emotion detection, analysis and visualization using COVID-19 tweets" proposes a deep learning pipeline to identify the emotion in a tweet. The paper also introduces a custom Q&A RoBERTa model to extract the primary words or phrase related to the identified emotion. To demonstrate the effectiveness and accuracy of the proposed models, this paper includes a historical emotion analysis during the COVID-19 pandemic in the USA.

Paper IV titled "A deep learning approach for ideology detection and polarization analysis during the COVID-19 pandemic leveraging social media." proposes a transformer-based deep learning model that detects the political ideology in the tweets. The model uses emotion in a tweet as a feature for ideology detection. Further, an emotion classification method is proposed in the paper leveraging the adversarial sample generation technique that improved the performance of the emotion detection significantly. A historical study of polarization on the topics "Masks" and "Vaccines" in the United States is also presented in this paper.

2. LITERATURE REVIEW

The scope of this dissertation primarily deals with effective disaster management using social media (Twitter) data. It is essential to acquire immediate information for an ongoing event and extract indirect information such as sentiment or emotion for future correlated events. Therefore, our research focused on collecting social media data in real-time, classifying relevant information, and extracting emotion during a crisis. To achieve our goals, we leverage machine learning tools and methods for predictive modeling. Therefore, we classify the related work into three categories: 1) Social media for disaster management, 2) Tweets classification, and 3) Emotion Detection.

2.1. SOCIAL MEDIA FOR DISASTER MANAGEMENT

The use of social media and networks for disaster management started to getting attention in the last decade. In 2021, Keim et al. [53] present a comparison between social networks and traditional media for information flow, adaptability, cost receptiveness, and timeliness of the data. The authors show the benefits of social media in spreading information and analysis of a situation. They discussed the new method of peer-to-peer data availability (which is social media communication) to others that do not need any central coordination. The authors also discussed the possibility of using social media for the communication in various situations due to the robusness of the platform. Gao et al. [25] present research showing the uses of social media during the catastrophic Haiti earthquake when mobile networks were unavailable. The authors claimed that the messages and photos shared on the microblogs by many individuals help the affected area to raise an enormous fund as it creates more appeal to donors. The authors adapted crowdsourcing for designing coordination protocols and mechanisms for the communication between the organizations and their relief activities. The developed crowdsourcing tool collects and analyzes the data from social media and allows relief organizations to obtain relief quickly and effortlessly. Imran et al. [41] use social media content and categorizes those based on location, seeking help, fundraising request, casualty reports, caution, and advice. The goal was to use social media information for effective response. Authors at [58] focus on Twitter as a tool for the emergency response organization to communicate with people and gain valuable information. The researchers investigated the use of social media in disaster management in two primary categories. First, the use of social media to coordinate with resources for rescue activities. Second, how the victims use social media to seek help and how the rescue organizations provide support. The emergency management professionals use the data obtained from social media to get detailed information using different tools. The authors explained the method using a case study and showed the benefit of social media for disaster response.

Palen et al. [86] analyzed the extensive use of Twitter data in case of mass convergence or disaster situations such as the Southern California Wildfires. Jie et al. [99] proposed a system that uses data mining and a machine learning mechanism to extract the data generated by Twitter messages during a crisis. Sifting the relevant data from the burst of social media messages is challenging. The authors explained the methods to address this issue using various data mining techniques. Recently, Yang et al. [97] propose a rescue scheduling algorithm on Hurricane Harvey. This algorithm connects the victims with the scattered volunteers. However, synchronize rescue efforts along with the government and organization might be more effective and faster. Although disaster management using social media became popular, there is still a lack of effective and efficient solutions and applications that address various challenges and facilitate different use cases.

2.2. TWEET CLASSIFICATION

Tweets classification is primarily a text classification task. There are many well-established examples for text classification and extracting information from unstructured text. Classifying document as the set of specified topics of interest and grouping document using clustering or topic modeling are two primary approaches to text classifying. The initial approach for text classification is to extract features from the text. Typical features include TF-IDF [79] of the bag of words. The Naive Bayes classifier is one of the popular classifiers where it models the documents distribution using probability. Another most widely used entity in classification is Support Vector Machines (SVM), which tries to draw a linear separator plane among the classes [5]. To perform the classification K-Nearest Neighbor Classifier [37] offer proximity-based classifier and use distance measurement among the words. KNN classifier assumes that the documents which belong to same class should have close properly such as cosine similarity measurement. Bootstrap aggregation method XGBoost [16] has shown good potential in text classification. XGBoost trains multiple classifiers with weak dependencies. Further, it aggregates the result from all the classifiers yielding a satiable performance.

Along with general machine learning algorithms, Neural Network provides some robust idea to classify text document. The idea of the deep neural network for natural language processing first use in [19]. The authors use a multitask learning model using the neural network. Conneau et al. [6] propose a deep neural network consisting 29 layers for natural language processing. The authors use a deep stack of local operation to learn the hierarchical representation of a sentence. Combining with external pre-learned word vector such as GloVe [77], a neural network can create a better classification model. Recently, transformers are increasingly popular to solve different natural language problems. Many researchers use or adopt transformers such as BERT (Bidirectional Encoder Representations from Transformers) [22], which produces contextualized word vectors that can be highly useful for different NLP tasks. In [29], the authors present a discussion comparing the performance of BERT against traditional text classification methods. Some other recent transformers such as RoBERTa [61], ALBERT [57], and XLNet [96] also showing promising performance for text classification. Though several well-performed text classification methods are available, most of those methods are not optimized for micro-text classification. Besides, tweets present some unique structure of the language representations which challenges existing text classification approaches.

2.3. TWEET EMOTION DETECTION

Tweet emotion detection is considered a text classification problem in most of the traditional tweet emotional classification [67, 44] approaches. The winners of the multi-label emotion classification task of SemEval-2018 Task1, Baziotis et al. [12] and Meisheri et al. [64] use bidirectional LSTM with the attention mechanism. Park et al. [74] try to classify the emotions by training two models and aggregating those. The authors use regularized linear regression and logistic regression classifier chains for emotion classification. In recent years researchers are considering tweet emotion classification as a unique problem compared to text classification. Tashtoush et al. [87] explores Tweets Emotion Prediction using Fuzzy Logic System to create a sentiment analysis system that extrapolates the text and emojis. The idea is to create a model that can acquire information individually from text and emojis in the tweet and use that information to improve the accuracy. Hasan et al. [38] introduce Skip-Thought, a deep learning model for emotion detection. The model generates a set of word vectors and embeds that with pre-trained sentences for emotion classification. Xiangsheng et al. [59] propose a novel model named HNN (Hybrid Neural Networks) for emotion detection. The authors use a pre-trained LSM (Latent Semantic Machine) for initial training and finally perform fine-tuning using a deep neural network. However, this hybrid model was inconsistent in the different training sessions and needs further improvements.

While there are several works available on tweet emotion detection, there are only a few attempts to classify the tweet emotion during the COVID-19 pandemic because of the lack of available datasets. None of the above works perform emotion classification on crisis datasets which might structurally different and represents emotions uniquely. Yang et al.

[95] introduce a COVID-19 dataset. The authors further use XLNet, AraBert, and ERNIE to detect emotion in English, Arabic, and Chinese language text. Imran et al. [40] use LSTM model to study cross-cultural polarity during COVID-19 pandemic. However, the study is limited to sentiment analysis only. Authors in [28] propose ESTeR , an unsupervised model for identifying emotions in the COVID-19 tweets. The proposed approach creates word graphs using a similarity function from the existing annotated data for scoring input texts with reference to a given set of emotions. The word graphs were obtained from the random walk considering co-occurrence of the words for emotion detection. While this approach outperforms the existing unsupervised method, it is still underperforming compared to the supervised models.

2.4. ADVERSARIAL SAMPLE GENERATION FOR EMOTION CLASSIFICATION

Adversarial learning approaches are widely used in image classification and segmentation problems. In recent years adversarial learning is also gaining popularity in Natural Language Processing (NLP) to solve complex problems. A team of researchers from Google and OpenAi, introduce adversarial training methods for semi-supervised text classification in [66]. The authors used perturbations in the text embedding with a Recurrent Neural Network (RNN) and outperformed the state-of-the-art results. Daniel et al. [33] explore domain adversarial training for low-resource text classification. The authors experimented with transfer learning from one language to another low-resource language using adversarial technique and showed the benefits. The authors expanded domain-adversarial neural network architecture to multi-source domains and evaluate the model performance to prove their claim. The authors also used pool-based active learning to achieve satisfactory results. Croce et al. [20] used Generative Adversarial Learning to improve the BERT [22] and extended it for the robust text classification. In the authors experiments, adopting adversarial training to enable semi-supervised learning in Transformer-based architectures improves the model performance with fewer labeled examples.

Although there is a scarcity of research works on adversarial learning for emotion classification from text, recently the idea is gaining popularity. In a recent work, Bo Peng et al. [76] proposed an adversarial learning method for sentiment word embedding in order to force a generator to create word embedding with high-quality utilizing the semantic and sentiment information. Authors in [93], utilize adversarial multi-task learning for Aggressive language detection (ALD) from tweets. The authors created a task discriminator for text normalization to improve aggressive language detection. The proposed adversarial framework uses the private and shared text encoder to learn the underlying common features across the labels and thus improve the performance. The authors proposed a confrontation network using transfer learning to achieve rapid theme classification from the text in [36]. The authors developed an adversarial network to extract the common features of different tasks which in turn improved the performance. The authors leveraged generative adversarial networks to combine several single tasks, called Joint-multi-kernel (MCMK) model.While adversarial learning and generative adversarial training methods are gaining traction in natural language processing, still a lot of research and experiments are critical to ensure a robust method for emotion detection using social media data.

2.5. POLARIZATION DETECTION USING TWITTER

Twitter became increasingly popular for public communication and information monitoring during a crisis. Twitter data analysis during the COVID-19 pandemic became a topic of interest because of different opinions, misinformation, and controversies. In [55] Ramez et al. presented their works on misinformation propagation and quantification, related to COVID-19 using tweets. The authors analyze the Twitter data to detect polarity, anxiety, and misinformation regarding medicine or medical methods which can be harmful. Researchers observed extensive social and political polarization in the social media content during the COVID-19 pandemic. Polarization detection in social media became very popular during the pandemic. Polarization is highly connected with political affiliation and thus political ideology detection plays a critical role to analyze and understand polar opinions. There are two primary approaches for polarization and political ideology detection in the tweets: content-based and network-based. In content-based approaches [2] user profile information such as metadata, location, race, gender, etc. are used along with the tweet text. Further, the processed information is compared with the seed users to infer the political affiliation of a given user. This method mostly considers the whole user profile to detect the ideology of a user instead of the ideology expressed in a single tweet. In the network-based approach, a network of similar ideology people is formed using the interaction between the people with the seed user. Authors in [46] built a user network using retweets, engagement, and followers. The authors further use the network to detect and analyze the echo chambers. While an interaction network mostly contains people with similar ideologies it can also contain people with opposite principles as they can interact to oppose the information or view. Polarization exploration for different topics such as "facial masks" and "vaccines" also became very popular during the COVID-19 pandemic. Yeung et al. [98] explore the polarization of personal face masks during the pandemic. The authors analyze the people with different demographic including but not limited to age, gender, geographic region, and household income. The authors performed valence-aware sentiment analysis for polarization detection. The authors took a content-based approach to detect the political affiliation using a set of topics of interest. Jiang et al. [47] analyzed the polarization of the COVID-19 vaccine using followers and expressions in the tweets. The authors explore the likelihood and hesitancy against vaccines among different political ideologies. Most of the related works determine the political ideology of a user and further deduced all of the tweets by that user exhibit a similar ideology. In this research, although we take a content-based approach to detect political ideology, however, instead of user interaction information or profile meta-information we only use the tweet text and the emotion in the text.

PAPER

I. A DEEP LEARNING APPROACH FOR TWEET CLASSIFICATION AND RESCUE SCHEDULING FOR EFFECTIVE DISASTER MANAGEMENT

Md Yasin Kabir and Sanjay Madria
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri 65401
Email: mkabir@mst.edu and madrias@mst.edu

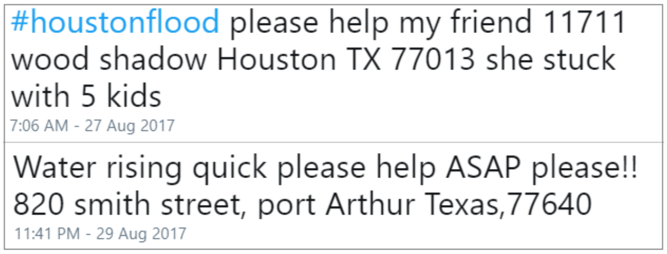
ABSTRACT

Every activity in disaster management demands accurate and up-to-date information to allow a quick, easy, and cost-effective response to reduce the possible loss of lives and properties. It is a challenging and complex task to acquire information from different regions of a disaster-affected area in a timely fashion. The extensive spread and reach of social media and networks such as Twitter allow people to share information in real-time. However, gathering of valuable information requires a series of operations such as (1) processing each tweet for the text classification, (2) possible location determination of people needing help based on tweets, and (3) priority calculations of rescue tasks based on the classification of tweets. These are three primary challenges in developing an effective rescue scheduling operation using social media data. In this paper, first, we propose a deep learning model combining attention based Bi-directional Long Short-Term Memory (BLSTM) and Convolutional Neural Network (CNN) to classify the tweets. Next, we perform feature engineering to create an auxiliary feature map which dramatically increases the model accuracy. In our experiments using data from Hurricanes Harvey and Irma, it is observed that our proposed approach performs better compared to other classification methods based on Precision, Recall, F1-score, and Accuracy, and is highly effective to determine the priority of a tweet. Furthermore, to evaluate the effectiveness and robustness of the proposed classification model a merged dataset comprises of 4 different datasets from CrisisNLP and another 15 different disasters data from CrisisLex are used. Finally, we develop an adaptive multi-task hybrid scheduling algorithm considering resource constraints to perform an effective rescue scheduling operation considering different rescue priorities.

Keywords: Deep Learning, Neural Network, Social Media, Disaster management, Rescue Scheduling, Priority Determination.

1. INTRODUCTION

Social media such as Twitter and Facebook experiencing mass adaptation and exponential growth. The roles of social media extended but not limited to health and disease analysis and propagation detection [1], Quantifying controversial information [2], and disaster crisis management [3, 4]. Natural disasters frequently disrupt regular communication due to the damaged infrastructures [5] which lead to an outflow of information. A report on Hurricane Sandy [6] shows that people were using social media more frequently to communicate. People were seeking help quickly and promptly as they strive to contact friends and family in and out of the disaster area, looking for information regarding transport, shelter, and food. Hence, The huge flow of information over social media can be beneficial in managing a natural disaster more effectively. During Hurricane Sandy, Twitter proved its usefulness, and at the time of Hurricane Harvey and Irma, again Twitter played a crucial role in the rescue, donation, and recovery. Figure 1 represents two tweets seeking rescue during Hurricane Harvey. People also tweeted similarly at the time of Hurricane Irma. However, while the use of social network seems appealing, still most of the applications are lacking features and fall short in their usability [7].



#houstonflood please help my friend 11711 wood shadow Houston TX 77013 she stuck with 5 kids

7:06 AM - 27 Aug 2017

Water rising quick please help ASAP please!! 820 smith street, port Arthur Texas,77640

11:41 PM - 29 Aug 2017

Figure 1. Examples of rescue requesting tweets

Institutional and Volunteer rescue efforts save a lot of lives during a crisis. However, those rescue missions are not well-organized and structured due to uncertainty. Individual volunteers have time constraints and lack of resources. Moreover, some rescue missions might need extra precaution, advanced equipment, and medical facilities. Besides that, due to the variety of help requesting tweets, some of those tweets might be out of sight. Hence, an automated system is essential to understand the context of the tweets, classify the specific tweets for rescue, prioritize those tweets based on context, and then schedule rescue missions and allocate necessary resources accordingly. Our primary contributions in this paper are:

- Developing a multi-headed binary classifier to classify the tweets into six different classes using deep learning where a single tweet can belong to multiple classes. We use a unique machine learning pipeline with a set of punctuation-based auxiliary features which are specifically correlated with the disaster-related tweets.

- Evaluating and comparing the proposed model with different machine learning models and diverse datasets.

- We formally introduce a method for priority determination of each rescue request which plays a crucial role in maintaining fairness in the rescue scheduling.

- We propose a resource constraint and burst time adaptive rescue scheduling algorithm with multi-tasking and priority balancing to perform improved rescue operations.

2. RELATED WORK

2.1. SOCIAL MEDIA FOR DISASTER MANAGEMENT

Most of the prior research work research works using social media and networks for disaster management are focused on assessing the disaster situation, and a little, if any, is focused on their use in rescue mission and planning. Authors in [4] proposed a rescue scheduling algorithm on Hurricane Harvey which connects the victims with the scattered volunteers. A heuristic multi-agent reinforcement learning scheduling algorithm, named as ResQ [8], utilizes reinforcement learning to coordinates the volunteers and the victims during a disaster. [9] proposed a system that uses machine learning mechanism to extract the data that is generated by Twitter messages during a crisis. Authors in [10] presented how social media communication was used during the catastrophic Haiti earthquake. They adapted the method of crowd-sourcing for designing coordination protocols and mechanisms in order to create coordination between the organizations and their relief activities. [11] analyzed the extensive use of Twitter data in case of mass convergence or disaster situation such as the Southern California Wildfire. After several devastating incidents, a few disaster management applications such as Ushahidi [12] have been developed.

2.2. TWEETS CLASSIFICATION

The basic approach for tweet classification is to extract features from the text. Support Vector Machines (SVM) is one of the widely used entity in classification, which draws a linear separator plane among the classes [13]. To perform the classification, K-Nearest Neighbor Classifier [14] offers proximity-based classifier, and uses distance measurement among the words.

The idea of the deep neural network for natural language processing first used in [15] uses a multitask learning model using the neural network. [16] proposed a deep neural network consisting of 29 layers for natural language processing. [17] and [18] showed that combining with external pre-learned word vectors such as GloVe [19], a neural network can be trained better for the disaster datasets. Our proposed deep learning model took inspiration form their work. However, those works did not consider any auxiliary features or attention layer. As a tweet has character length restriction, attention layer with domain-specific engineered auxiliary features can be highly influential. In this work, we create a set of auxiliary features and use an attention based deep neural network to classify the tweets into 6 different classes where each class represents a binary output label, and a single tweet can belong to multiple classes.

2.3. SCHEDULING ALGORITHMS

The scheduling algorithms intend to optimize the time and the use of resources among different parties employing certain constraints. The primary purpose of a scheduling algorithm is to ensure fairness among the participants while maximizing resource utilization. First-Come-First-Served (FCFS) algorithm can not provide fairness when someone cannot wait to use the resource or when someone needs a priority based on a situation. [20] worked with fixed-priority scheduling to consider the complexity of determining whether a set of periodic real-time tasks can be scheduled on $m > 1$. [21] proposed fixed-priority scheduling using a fixed-relative deadline. After a certain period of time, a task became suspended upon failure and the resource became available. [22] presents a scheduling algorithm for emergency medical rescue conflict monitoring and dispatch scheduling based on the hybrid estimation and intent inference. [23] took a heuristic approach for solving the rescue unit assignment and scheduling problem under the resource constraints. In [4], the authors discuss the utilization of the public resources for disaster rescue with the priority based scheduling policy. The authors present a discussion about the fairness and importance of priority based on rescue scheduling. However, there is no formulation to determine the priority scores of rescue scheduling tasks. In this paper, we formally define a method to determine the priority score of rescue tasks and propose a multi-task hybrid scheduling policy using priority, based on certain criteria to develop an effective and efficient rescue scheduling algorithm.

3. TWEETS CLASSIFICATION AND INFORMATION RETRIEVAL

Twitter data from two different natural disasters (Hurricane Harvey and Hurricane Irma) were collected for this work. We collected these tweets from August 26 to August 31, 2017 and September 10 to September 17, 2017, respectively. We use Twitter Stream API to collect the tweets along with various meta-information such as user information, geo-location, tags, entities, etc. The pre-processing step involves discarding non-English tweets, filtering noises and duplicates, removing special characters, stop-words, and jargons.

According to the FEMA, WHO, and NCDP, the "vulnerable populations" or "at-risk individuals" includes children, senior citizens, pregnant women, disabled, sick or injured persons. A tweet classifier is developed using the neural network to identify whether a tweet falls into one or more classes from six different classes (Rescue needed, DECW, Water needed, Injured, Sick, Flood). DECW stands for Disabled, Elderly, Children and Women. Those six classes help in determining the rescue situation, and their priorities along with the resources needed or requested in a tweet. We use the label Water_needed as a request for drinkable water identified as a vital resource during any disaster or emergency by the Centers for Disease Control and Prevention (CDC).

3.1. DEEP NEURAL NETWORK

The proposed deep learning model comprises 7 primary components. Figure 2 depicts the fundamental system architecture of the model.

Input layer: Pre-processed tweets fed to the input layer which is connected with the embedding layer.

Embedding layer: This layer encodes the input into real-valued vectors using lookup tables. In this work, we used a pretrained word vectors named Crisis [17] and GloVe [19] which generates a feature word vectors using co-occurrences based statistical model. Embedding applied to the words aids to map all tokenized words in every tweet to their respective word vector tables. To unify the feature vector matrix, appropriate padding is added.

BLSTM layer: The Long-Short Term Memory (LSTM) is a specialized version of Recurrent Neural Network (RNN) that is capable of learning long term dependencies. While LSTM can only see and learn from past input data, Bidirectional LSTM (BLSTM) runs input in both forward and backward direction. This bidirectional feature of BLSTM is critical for the various applications involved with understanding complex language [24].

The input gate i_t, forget gate f_t, output gate o_t, and cell state activation c_t of the implemented LSTM version in this work can be defined by the equations (1)-(5) where σ represents the logistic sigmoid function, h represents the respective hidden vectors, and W is the weight matrix. A detailed explanation of each equation and more about LSTM are available on [25].

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{2}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \tag{3}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{4}$$

$$h_t = o_t \tanh\left(c_t\right) \tag{5}$$

Attention layer: Every word in a sentence does not contribute equally to represent the semantic meaning and the primary concept of attention [26] originated from this observation. We use a word-level deterministic, differentiable attention mechanism to identify

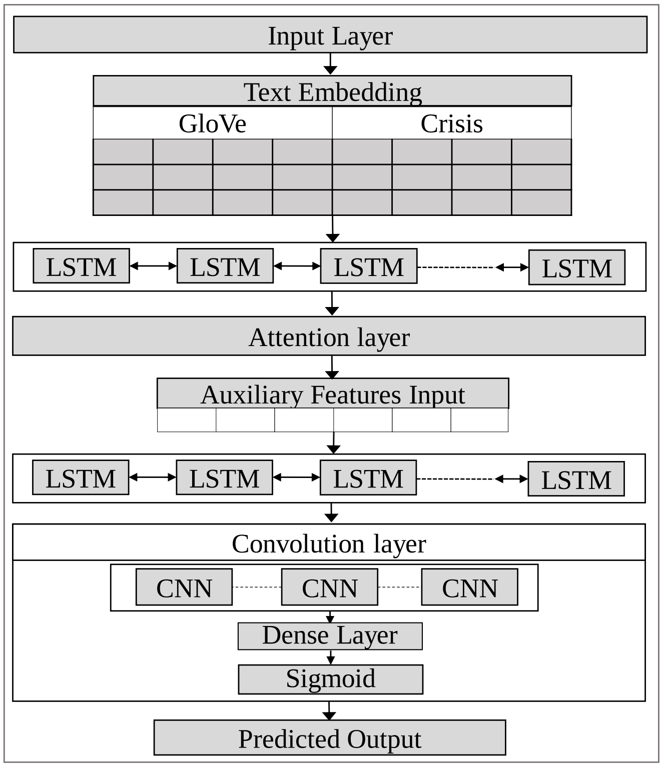


Figure 2. System architecture of the proposed model

the words with the closer semantic relationship in a tweet. Equation 6 represents the attention score ei, e of each word t in a sentence i, where g is an activation function. More information on the attention mechanism is available on [27].

$$e_{i,j} = g\left(W h_t c\right) \tag{6}$$

Auxiliary features input: A tweet can only contain 280 characters (previously 140) which forces a user to express emotions in a different way compared to a traditional English sentence. People use extra punctuations and emoticons to intensify the meaning of a tweet. We also observed (e.g. Figure 1) greater use of numeric characters in a rescue seeking tweet due to the fact that people try to share location in the tweets In this work, we perform feature engineering to obtain a set of specific auxiliary features that can assist the classification

Table 1. Auxiliary Features

polarity, subjectivity, sentiment, wordsVsLength, exclamationMarks,
digitVsLength, punctuationVsLength, nounsVsWwords, sadVsWords,
capitalsVsWords, uniqueWords, numberOfTags.

model to learn better. A list of extracted auxiliary features that shows noticeable influence during the model evaluation is given in Table 4. The well-known Natural Language Toolkit (NLTK) is used to extract those features.

Convolution layer: The convolution layer performs a matrix-vector operation in the sentence-level representation sequence. Let us assume that $H \in \mathbb{R}^{d*w}$ be the weight matrix, and the feature mapping done as $c \in \mathbb{R}^{l-w+1}$. The i-th element of the feature map can be defined as:

$$c_i = \sigma\left(\sum \left(C\left[*, i : i+w\right] oH\right) + b\right) \tag{7}$$

In sentence-level representation, $C[*, i : i+w]$ is the i-th to i+w-th column vector. The word vectors pass through the convolution layers [28] where all the input information merged together to produce a features map. The Rectified Linear Unit (ReLU) used as the activation to deal with the non-linearity in the convolution layer and generate a rectified feature map. Finally, the dense layers are activated for generating the outputs.

Output layer: The activation function *sigmoid* is used in the dense layer as we want to perform multi headed binary classification. The model produces binary values for all six target output classes. Detailed information on model hyperparameters and evaluation results is given in Section 5.1.

3.2. LOCATION EXTRACTION

Due to the privacy policy of Twitter, most of the tweets do not contain any location information. In those cases, we try to extract location using user profile meta information and the location information provided in the tweeted text. Combining the Stanford Named Entity Recognizer (NER) [29] and Google map API, an application is built for extracting location.

4. RESCUE SCHEDULING

4.1. PROBLEM SPECIFICATION

Let us assume that the number of rescue teams be m with n pending rescue tasks. Let the processing time of rescue task j by team i be t_{ij}, where $1 \leq i \leq m, 1 \leq j \leq n$. Based on a typical disaster situation, we consider that the number of rescue tasks is greater than or equal to the number of rescue teams ($n \geq m$). The problem is to organize and assign the tasks to rescue teams in such a way that the amount of waiting time for each rescue mission is minimized. However, due to the inconsistent nature of the rescue tasks and the location of the incidents, the formulation of this problem faces the following major challenges.

1. Depending on the capabilities and resources every rescue team may not capable of processing each task.

2. It is difficult to precisely estimate the required time t_{ij} for a task due to the uncertainty of the environment and location of an incident.

3. Tasks might have different priorities based on the people needed to execute them and their physical condition. The environmental condition of a person such as surrounded by flood water, or fire should also be taken into account while determining priority.

Along with the above challenges, we also consider the following restrictions and conditions to formulate the problem effectively.

- We imposed a time t_j for a task j, where t_j denotes the required time for a rescue team i to move from initial rescue center to the place of incident. The time for moving from the location of a task j to another task j' is represented by $t_{jj'}$.
- Every team requires a preparation time before leaving for a scheduled rescue job from their respective rescue management station. The preparation time is denoted as t_i for every team for a specific task j. Also, after a certain period, every team might require a resting time of t_{ir} before the next task.

Considering the above sequence of times (t_{ij}, t_j, $t_{jj'}$, t_i, and t_{ir}), we can estimate a probable time for a rescue mission. Although the time can be changed based on the situation, we consider some constant time variable considering the distance of a task location and the probable situation of the environment around the incident.

4.2. PRIORITY DETERMINATION

A significant step for the rescue scheduling algorithm is determining the priority of rescue tasks. We use the output labels of tweet classifier (Section 3.1) and assign a weight for each label to determine the priority of that tweet. Assume the assigned weights for different labels of the tweets is represented by a vector $w_j = [w_1, w_2, ...w_n]$. A feature vector $\alpha_i = [\alpha_1, \alpha_2, ...\alpha_m]$ also used which denotes the weight of other considerable variables such as the number of victims, real-time environmental conditions and future weather forecasts of a specific location. Equation 1 represents the formula to estimate the priority for a rescue task. The base priority value of a tweet is 1 where the maximum priority score can be 10.

$$f_p = \sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{n} w_j \tag{8}$$

4.3. RESCUE SCHEDULING ALGORITHMS

General scheduling algorithms are not applicable in disaster rescue scenario as those algorithms might be unfair due to different situations, physical conditions, and the critical importance of human life. A priority-based scheduling algorithm might provide a better solution where we need to consider and determine the priority continuously. In a disaster scenario, priorities can change with time and environmental conditions. Hence, We develop an effective rescue scheduling algorithm considering priority, environmental severity, and processing time of every single task. We like to define the terms which we use to represent our algorithms.

- Tasks: A task is the combination of one or more valid rescue requests by an individual or multiple people. A list of valid requests forms a sequence of tasks which demands to be scheduled appropriately.

- Processors: The number of rescue units which can complete a given task is the processors. A processor is responsible to execute a given task, release the resources upon completion, and get back to the initial state to execute a new task.

- Arrival Time: The time of receiving a valid rescue request represents the arrival time for a specific task. In our rescue scheduling system, arrival is the time-stamp of a tweet.

- Burst Time: The probable time required to complete a task by a processor can be defined as burst time. The burst time is realistically represents the service time of a processor for a rescue mission. In a disaster scenario, estimating appropriate burst time is very challenging. Similar tasks might take different times to complete under separate circumstances. To address this issue, first, we assume a probable burst time based on the rescue operations in previous disasters. After the completion of a few rescue missions, the burst time of the future mission is determined using the actual completion time of those missions. To predict the future burst time, we use the exponential averaging method. Given n tasks (taskSeq[1...n]) and burst time for tasks t_i, the predicted burst time for the next task $taskSeq_n + 1$ will be:

$$BT_{n+1} = \alpha T_n + (1 - \alpha) BT_n \tag{9}$$

In the above equation, α is a constant factor ranging (0 <= α <= 1). The value that can predict the best possible burst time will be assigned as α. The variable BT_n denotes the predicted or assumed burst time for the task n, and T_n represents the actual burst time needed for completing task n.

Three different scheduling algorithms are implemented for the experiments. All of those algorithms are implemented using multiple processors as it is expected to have more than one rescue unit in an emergency rescue situation. Although we emphasize on Multi-task Hybrid Scheduling algorithm, however, we study fundamental rescue algorithms to understand the limitations of these established methods. This study also indicates the necessity of a novel adaptive Hybrid Scheduling algorithm for a disaster scenario.

4.3.1. First-Come-First-Serve (FCFS).

In FCFS scheduling system, the task requests are sequentially processed in the order of the arrival time. A sequence of tasks list (taskSeq) with the requests arrival time (arrivalTime) and probable burst time (burstTime) is fed to the algorithm as input. The algorithm returns the scheduled tasks sequence with the possible start time. However, estimate burst time can change and needed to update while the processor is processing a task. While FCFS is a simplest scheduling algorithm, it has two major concerns which need some attention.

- In a disaster scenario, every rescue request is not similarly critical. FCFS fails to consider the tasks which have an urgency of completion.
- FCFS is a non-preemptive scheduling algorithm which is responsible for the short jobs to wait longer based on the sequence order.

4.3.2. Priority Scheduling. In a disaster scenario, conducting rescue missions based on priority can be crucial. There can be rescue requests which can wait longer, and might not be critical like other requests. A priority-based scheduling algorithm is more appropriate considering those facts. The algorithm executes the task using an ordered queue with high to low priority. A priority queue based scheduling algorithm is demonstrated in the Algorithm 1.

4.3.3. Multi-Task Hybrid Scheduling. The incidents at the end of the priority queue need to wait longer when there is a large scale disaster because of plenty of rescue requests. Assume there are some tasks which need to wait longer for rescue due to lower priority. Suppose some of those tasks are located in an area where the disaster situation is worsening by time. The severity can increase fast at those places. A priority balancing scheduling policy might be helpful in such a scenario. It may need more information and human input to decide how and when to increase the priority of a task before it enters into critical condition. To solve this dilemma, we introduce a priority balancing module which re-calculate the priority score after the completion of each rescue mission.

Instead of a single rescue task in a mission, a rescue team can execute multiple tasks depending on available resources. For example, in a flood situation, several individuals can be rescued in the same boat and transferred to a shelter together. We illustrate this idea along with priority balancing in Algorithm 2. A processor can be assigned for multiple tasks in a single rescue mission if it has available resources. We use a 2 miles radius area for this purpose. A processor looks for other available tasks which are within 2 miles radius of the assigned event. It will incorporate multiple tasks as long as the processor has adequate resources and executes those tasks sequentially using priorities. Comparative performance evaluation of the algorithms is present in Section 5.2. In Section 5.3 we describe and demonstrate the Multi-Task Hybrid Scheduling algorithm using a real-world disaster scenario.

Algorithm 1 Priority scheduling with multi-processors

```
Input: processorNo, taskSeq$[1...n]$, arrivalTime$[at_1...at_n]$, burstTime$[bt_1....bt_n]$,
    tasksPriority$[1...n]$;
Output: scheduleSeq$[task_i...task_n]$, startTime$[st_1...st_n]$,
    turnAroundTime, avgWaitingTime, avgTurnAroundTime;
    Initialization: All the processors K are released and ready to begin a task.
    Initialize, scheduleSeq, startTime, and turnAroundTime as list; currTime = 0, waitingTime = 0, totalTurnAroundTime = 0;
    Sort the taskSeq, arrivalTime, burstTime using taskPriority and assign the tasks in priority queue $P_{queue}$;
if (new task request) then
    update $P_{queue}$, taskSeq, arrivalTime, burstTime, number of tasks n;
end if
for $i$ = 1 to n do
    select task i to be processed;
    dequeue the root element from $P_{queue}$
    scheduleSeq.append(i);
    $K^*$ are the available processors to process task i;
    if ($K^* \neq \emptyset$) then
        assign current task to $K$;
        if (currTime<arrivalTime[i]) then
            currTime = arrivalTime[i];
        end if
        startTime.append(currTime);
        waitingTime = waitingTime + (currTime-arrivalTime[i]);
        completionTime = currTime + burstTime[i];
        currentTrunAroundTime = completionTime - arrivalTime[i];
        totalTurnAroundTime = totalTurnAroundTime + currentTrunAroundTime;
        turnAroundTime.append(currentTrunAroundTime);
        release $K$;
    else
        return to if
    end if
end for
calculate avgWaitingTime, avgTurnAroundTime;
```

Algorithm 2 Multi-tasks Hybrid Scheduling

```
Input: processorNo, taskSeq$[1...n]$, tasksPriority$[1...n]$, arrivalTime$[at_1...at_n]$,
    burstTime$[bt_1....bt_n]$, taskslocation[1...n], disRadius;
Output: scheduleSeq$[task_i...task_n]$, startTime$[st_1...st_n]$,
    turnAroundTime, avgWaitingTime, avgTurnAroundTime;
    Initialization: All the processors K are released and ready to begin a task.
    Initialize, scheduleSeq, startTime, and turnAroundTime as; currTime = 0, waitingTime
    = 0, totalTurnAroundTime = 0;
    Sort the variables in descending order using taskPriority. Re-sort the values in ascending
    order using burstTime and arrivalTime for same taskPriority tasks. Assign the tasks in
    priority queue $P_{queue}$;
if (new task request) then
    update $P_{queue}$, and resort taskSeq, arrivalTime, burstTime, number of tasks n;
end if
for $i$ = 1 to n do
    select task i to be processed;
    dequeue the root element from $P_{queue}$
    scheduleSeq.append(i);
    $K^*$ are the available processors to process task i;
    if ($K^* \neq \emptyset$ and available $K$ is capable of addressing task $i$) then
        assign current task to $K$;
        for $m = i + 1$ to n do
            calculate the distance d of taskSeq[m] from current task using tasksLocation[m];
            if (d<disRadius and $K$ has the extra resources to complete taskSeq[m] after
            current task) then
                add taskSeq[m] with the current task queue and create a sub-scheduling for
                those tasks;
                dequeue the taskSeq[m] and update $P_{queue}$;
            end if
        end for
        estimate startTime, waitingTime, totalTurnAroundTime following the similar process
        of algorithm 2.
        release $K$;
    else
        return to if
    end if
end for
calculate avgWaitingTime, avgTurnAroundTime;
```

5. EXPERIMENTAL RESULTS

5.1. TWEETS CLASSIFIER EVALUATION

The primary goal of the tweet classification is to identify the people who need help and determine a priority score for each tweet based on the classified labels. To accomplish this goal, 4900 tweets were manually labeled into six different binary classes from 68,574 preprocessed tweets on Hurricane Harvey and Irma. We evaluate the proposed classification model on this labeled dataset and compared it with the well-established Logistic Regression (LR), Support Vector Machine (SVM) and fundamental CNN model. Moreover, in order to fully understand the effectiveness of our approach, we evaluate our model on several past disaster datasets obtained from CrisisNLP [17] and CrisisLex [30]. We use the same datasets and data settings of Nguyen et al.[18] and compare the output of our proposed model with the stated results of LR, SVM, and CNN in the same paper. To evaluate the robustness of our proposed technique, we merged 15 different disasters data from CrisisLex [30] and perform a binary classification which identifies the tweets relevant to a disaster.

Table 2. Hyperparameter values

Hyperparameter	**Value/Description**
Text embedding	Dimension: 300
BLSTM Layer	2 layers; 300 hidden units in each (Forward and Backward)
Conv1D Layer	3 layers; 300 convolution filters
Dense Layer	3 layers; First 2 layers have 150 and 75 units respectively and the last one is output (Dense)
Drop-out rate	Word Embedding: 0.3; Dense layer: 0.2 each;
Activation function	Conv1D, BLSTM, Dense: ReLU; Output Dense layer: Sigmoid;
Adam optimizer	Learning rate = 0.0001; $beta_1$=0.9;
Epochs and batch	Epochs = 10 to 25; batch size = 128;

5.1.1. Model Parameters. A set of optimal parameters is crucial to achieve desired performance results. We perform rigorous parameter tuning and select an optimal set that is used in all the experiments. We use the same parameter for better evaluation and model reproducibility. The popular evaluation metrics such as precision, recall, F1-score, accuracy, and AUC score is used to validate and compare the experimental result of the models. Table 2 represents the parameters used for the model.

5.1.2. Evaluation on Hurricane Harvey and Hurricane Irma Data. We use 4900 manually labeled tweets for this evaluation where 3920 tweets (80%) used for training and the rest of the 20% tweets used for testing. In the evaluation tables, we denote our model as CNN_{AAf}, which stands for CNN with Attention and Auxiliary features. We compare our model (CNN_{AAf}) with LR, SVM, and CNN without attention and auxiliary features. Our model outperformed all other models by more than 5% inaccuracy metrics. In terms of precision, the proposed model performed surprisingly well and outperformed the closed result of SVM by around 25%.

Table 3. Classifier evaluation (Hurricane Harvery and Irma)

Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
LR	55.8	93.0	69.7	84.5
SVM	65.1	85.4	73.9	88.5
CNN	61.6	90.8	73.4	87.5
CNN_{AAf}	81.7	93.4	87.2	93.7

Table 3 represents the full evaluation results for the different classifiers. Table 4 represents the evaluation metrics for individual classes (Hurricane Harvey and Irma) using CNN_{AAf} model. The distributions of the six classes in the data are Help - 29.1%, Flood - 26.3%, Water Needed - 4.9%, DCEW - 4.1%, Injured - 0.3%, Sick - 0.3%. However, we discarded labels Injured and Sick due to lack of enough data instances for training and testing so that it cannot influence the metrics of the model. As there are few true positive instances, those two labels achieve a higher rate of Accuracy although the model is not identifying true positive instances. We can also observe a better precision and accuracy for labels Water Needed and DCEW. This is happening as there are also a few true positive instances. However, still, the model has performed well for the recall and F1-score as the words found in the tweets for those labels have fewer variations.

Table 4. Evaluation metrics for individual classes

Class	**Precision**	**Recall**	**F1-score**	**Accuracy**
Help	87.9	97.7	91.2	94.9
Flood	78.2	94.1	85.3	91.3
Water Needed	87.5	71.4	78.7	98.0
DCEW	93.7	73.2	82.3	98.5
Weighted Avg	81.7	93.4	87.2	93.7

5.1.3. Evaluation on CrisisNLP and CrisisLex Datasets. We use the same datasets and class distributions consisting of Nepal Earthquake, California Earthquake, Typhoon Hagupit, and Cyclone PAM which is described in [17]. In that paper, the authors evaluate the models on event data, out-of-event data and a combination of both datasets. In Table 5, we represent the results on the combination of both datasets. Clearly, our proposed CNN_{AAf} model outperformed all other models in term of AUC score which the authors also used in the referenced paper. Auxiliary features have a high impact to better understand the semantic meaning of the tweets which is reflected on the AUC score.

Table 5. Classifier evaluation AUC scores (CrisisNLP)

Disaster Name	**LR**	**SVM**	CNN_I	CNN_{AAf}
Nepal Earthquake	82.6	83.6	84.8	87.5
California Earthquake	75.5	74.7	78.3	83.6
Typhoon Hagupit	75.9	77.64	85.8	88.3
Cyclone PAM	90.6	90.74	92.6	92.6

We consider 15 different natural disaster datasets from CrisisLex [30]. After removing null values and preprocessing the merged datasets contains 13738 data instances. We use around 75% data for training (9268) and validation (1030) and 25% data for testing (3440). The comparative evaluation result using sklearn metrics is presented in Table 6. It is observable that the domain-specific auxiliary features along with attention layer is highly beneficial for understanding and identifying crisis tweets. Our proposed approach can be used on a diverse set of datasets with good outcome and this might play a crucial role to develop quick response application on the disaster domain. More information on the 15 datasets is available in the extended version of our paper [31].

Table 6. Classifier evaluation (CrisisLex)

Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
LR	85.8	71.1	77.8	85.8
SVM	90.9	74.7	82.1	73.2
CNN	93.4	76.3	84.2	76.4
CNN_{AAf}	93.6	93.7	93.4	93.6

5.2. COMPUTATIONAL EXPERIMENT ON SCHEDULING

A computational experiment has been performed on the proposed algorithm in Section 4. For the purpose of evaluation and comparison, a data-set consisting of hurricane Harvey tweets between 27^{th} August 2017 and 31^{st} August 2017 have been used. To identify the rescue seeking tweets, the proposed tweet classification model is used. We processed the identified tweets to extract and determine the required information for the scheduling algorithm such as location, possible service time (burst time), and priority using the described process in Section 3. The priority of each tweet was determined on a scale of 10 using four classes (Flood, Water Needed, DCEW, and Sick or Injured), labeled by the classifier following Equation 1. The weights for those classes were assigned as 1.5, 1.5, 2 and 2.5, respectively. For the environmental feature vector, we use a random distribution between

0.5 to 2.5. However, automatic weight determination still remains an open problem for the research. Next, the probable service time was estimated for each of the rescue tasks. We use the normal distribution of average service time as 54 minutes which is described in [4]. Finally, after all the processing, 174 rescue seeking tweets were found from around 72 hours data frame. This sample size is relatively small and distributed over a longer period. Hence, we performed upsampling using resample and linear interpolation methods from python pandas library and created a dataset containing 550 rescue tasks to evaluate the rescue algorithms.

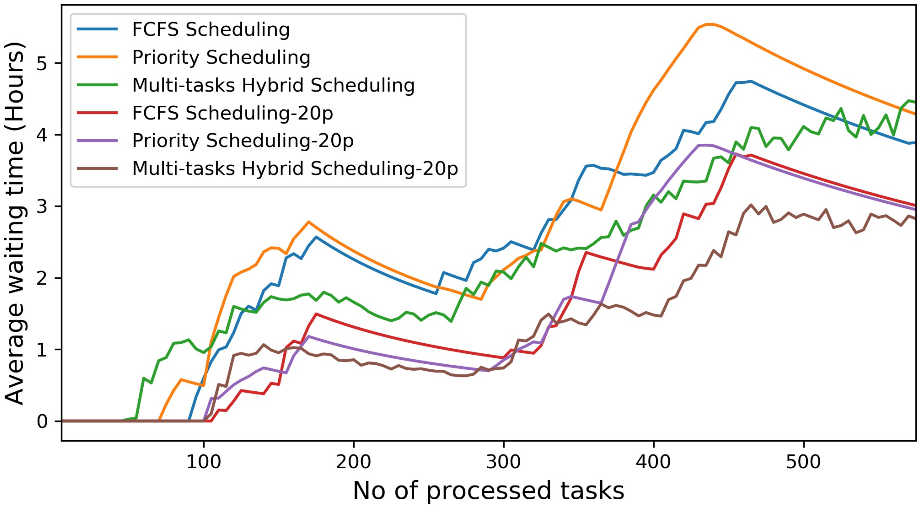


Figure 3. Average waiting time using 10 and 20 processors

The algorithms were implemented using the multiprocessing system. We use the number of rescue units (processors) as 10 and 20 to evaluate the performance of the scheduling algorithms. In Multi-task hybrid scheduling algorithm, the traveling time from one rescue location to another also considered while combining multiple tasks. Eventually, this estimation reduces the processing time for those tasks. Table 7 describes the summary of the three algorithms. In the table, 10p and 20p represent the number of processors used to execute those algorithms. The average waiting times are lowest in case of Multi-task hybrid scheduling algorithm. The average waiting time (hours) with the number of processed tasks is represented in Figure 3. The experimental results can be summarized as follows.

Table 7. Average waiting time summary

Algorithms	**Max avg WT**		**Mean avg WT**	
	10p	**20p**	**10p**	**20p**
FCFS	4.74	3.73	2.53	1.61
Priority	5.54	3.85	2.81	1.63
Multi-tasks Hybrid	4.47	3.02	2.24	1.31

- FCFS scheduling algorithm performs better comparing to Priority scheduling algorithm. However, in a disaster scenario, FCFS is not a fair policy to distribute the resources and rescue mission. Priority scheduling has a longer average waiting time because the lower priority tasks are waiting longer in the queue.
- Multi-tasks hybrid scheduling beats all other algorithm with respect to average waiting time. This algorithm is more practical for effective rescue scheduling and resource allocation as it consider resource constraints. It allows completing small tasks together of a nearest distance. Furthermore, it can be utilized to transfer the required resources (such as water, medicine) to the different locations while optimizing the average waiting time. However, the maximum average waiting time for this algorithm can be high for a task with less priority and larger processing time. It can happen when the location of a mission is far away with a low priority score.

5.3. EXPERIMENTAL ANALYSIS ON REAL-WORLD SCENARIO

A sample data-set is processed from the tweets during Hurricane Harvey to demonstrate Multi-task Hybrid Scheduling algorithm. An area of 20 square miles radius at Port Arthur, Texas has been selected for performing rescue operations. Figure 4 represents the geographical locations of the victims (Red icons) and the hyphothetical rescue operation base (Home icon) in the Port Arthur, Texas during hurricane Harvey. The ArcGIS javascript API [32] is used to create Figure 4 and 5. To demonstrate the algorithm, we assume that there is a rescue operation base at Tyrrell Elementary School, Port Arthur, TX. The experimental process can be summarized by the following steps:

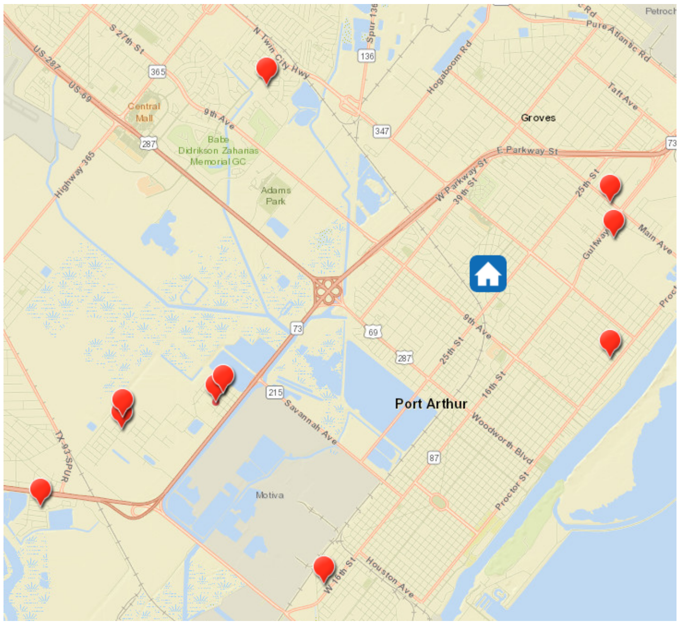


Figure 4. The positions of the victims and operation base.

- First, We have selected the rescue seeking tweets and extracted the location using the Stanford Named Entity Recognizer (NER) [29] and Google map API.
- Second, we extracted 10 tweets which were arrived first and located around 20 miles radius of the rescue operation base after 12pm of 30th August 2017.
- Third, the priority score, probable burst time and distance metrics have been calculated for each of the 10 rescue tasks.
- Finally, the Multi-task Hybrid Scheduling algorithm created the rescue schedule. We have simulated the experiment using 2 and 4 rescue units and two different distributions of the possible burst time. First, we assumed the required burst time to

be 54 minutes for each task based on the paper on hurricane Harvey rescue by Yang et al. [4]. Further, we use a random completion time for the first 5 tasks and predict the burst time of future rescue missions using equation 9.

Table 8. Classified tweet labels for priority determination

id	Flood	Water Needed	DCEW	Sick or Injured
1	1	1	0	1
2	1	0	0	0
3	0	1	1	0
4	1	0	0	1
5	0	0	0	0

Tables 8 and 9 represent example data sample of tweet labels and environmental features for priority calculation using Equation 1. We have used demo weights for the labels and environmental features as (Flood - 1.5, Water Needed - 1.5, DCEW - 2, Sick or Injured - 2.5, Storm - 1, Road Damaged - 1, forecasted storm - 0.5, forecasted flood - 0.5) respectively. We used experimental weights because determining the weights for those labels and features requires domain expert and extensive study. An appropriate authority or domain expert will be able to input precise weight values for the labels and environmental features considering the situation during an actual disaster. In the tables, id represent the respective tweet which is later refers to the same numbered $taskSeq$ in Table 10. The calculated priorities also presented in Table 10 as Priority Score.

Table 9. Environmental features example

id	Current		Forecasted	
	Storm	Road Damaged	Storm	Flood
1	0	1	1	0
2	0	0	1	0
3	0	1	0	1
4	0	1	0	0
5	1	0	0	0

Table 10 represents some columns of the processed sample data set of Port Arthur for the rescue scheduling. In the table, the burstTime is represented in minutes and distanceFromBase is measured in miles. To use Multi-task Hybrid Scheduling algorithm on the data, we need to assume some parameters. We consider the starting time of rescue mission as 14:00, the speed of the used vehicles or boats to rescue is 20MPH, and after the completion of each rescue mission a rescue unit requires 30 minutes as a preparation time before next task.

Table 10. Real-world data sample for simulation

taskSeq	**Arrival Time**	**Burst Time**	**Priority Score**	**Distance from Base**
1	12:13	54	7	5.1
2	12:45	54	2	5.0
3	12:58	54	5	6.9
4	14:07	54	5	7.0
5	14:46	54	1	3.9
6	15:23	75	2	4.5
7	16:10	70	8	1.9
8	16:52	30	7	7.7
9	17:30	35	5	1.8
10	18:05	45	6	2.0

The Multi-task Hybrid Scheduling algorithm can be demonstrated on the data in the Table 10 as follows. We use 2 rescue units to illustrate the algorithm.

1. The Start time of the rescue operation is 14:00. So, there will be 3 tasks in the queue at the time of the first iteration. The algorithm will first sort the tasks based on the priority score. Hence, the sorted sequence will be 1 >= 3 >= 2.
2. The location of the highest priority task (taskSeq 1) will be the point of interest. The algorithm will consider a perimeter of 2 square miles of that point and check if any other rescue task is there which can be combined. We can observe that taskSeq 1,2,

and 3 are within 2 miles radius. If a rescue unit contains enough resource for running those 3 operations sequentially, it will combine those tasks and rescue the people in a single go without coming back and forth to the base.

3. The algorithm will further create a sub-schedule of 3 tasks assigned to rescue unit 1. Task 1 has the highest priority and hence, the rescue unit will first go to location 1. From Figure 5, we can observe that tasks 1 and 3 are in a close distance. However, task 2 has a higher priority. As the algorithm emphasizes the priority score most, it will schedule task 2 before task 3. The rescue unit will assist the people in location 2 and then come back to location 3. Finally, it will come back to the base after the completion of all 3 tasks.

4. If there are multiple tasks with the high priority ($priority >= 7$), separate rescue unit will be assigned despite of there occurrence in a close proximity. In our experimental setup if two tasks with high priority are within 2 miles radius, the algorithm assigns two separate units for those two tasks. However, if there is only one rescue unit available, the algorithm will follow the above approach. Multiple tasks with the same priority will be sorted based on burst time and arrival time, respectively. Multiple tasks with same priority and burst time will be sorted using arrival time.

5. Based on the conditions, taskSeq 1, 2 and 3 will be assigned to rescue unit 1. Rescue Unit 2 will take care of taskSeq 4 which arrives at 14:07. The algorithm will wait until the completion of a task, after which a rescue unit became available.

6. The taskSeq 4 will complete first and rescue unit 2 will become available around 16:13. The algorithm will iterate again and sort the remaining tasks. At this point, the queue contains 3 tasks (taskSeq 5,6 and 7).

7. Employing the conditions, the sorted order for the tasks will be 7 >= 6 >= 5. The taskSeq 7 has a high priority and there are no other victims nearby. Hence, rescue unit 2 will be assigned to complete task 7.

8. Rescue unit 1 will be available again at 17:55. The algorithm will continue iterating until all of the 10 tasks are completed.

Table 11. Rescue scheduling output table of Multi-tasks Hybrid Scheduling algorithm using 2 rescue units

taskSeq	**Start Time**	**Route Distance**	**Route Duration**	**Waiting Time**	**TAround Time**	**Unit**
1	14:00	5.1	15	122	176	1
3	15:09	2.0	06	137	191	1
2	16:09	2.2	07	211	265	1
4	14:07	7.0	21	21	75	2
7	16:13	1.9	06	09	79	2
8	17:55	7.7	23	86	116	1
10	18:05	2.0	06	06	51	2
9	19:32	1.8	06	128	163	2
6	19:41	4.5	14	272	347	1
5	20:49	3.9	12	375	429	2

Table 11 represents some output values and rescue schedule for the data illustrated in Table 10. The column *StartTime* represents the scheduled time for the respective task. *RouteDistance* denotes the actual one-way path that a rescue unit needs to travel for a particular rescue mission. When multiple tasks are group together for a single mission the *RouteDistance* became the path between previous task and current task. For example, in Table 10, the distance of rescue location of taskSeq 3 from base is 6.9 miles. However, as tasks 1,2 and 3 grouped together the distance between the previous task 1 and task 3 became 2 miles. *RouteDuration* is the rounded time in minutes to travel the specific *RouteDistance*. In our experiment, we assume that a rescue unit needs 3 minutes to travel a mile. *WaitingTime* is the subtraction of *StartTime* and *ArrivalTime* with the addition of required travel time ($RouteDuration$) for a rescue location. The turnaround time is represented by $TAroundTime$ in the table which is the summation of $WaitingTime$ and $BurstTime$. The rightmost column in the table represents the assigned rescue unit for a task. After returning from a rescue mission to the base, a rescue unit requires a preparation time to become available for the next mission. In the experiment above, the rescue unit 1 reached at the base at 17:25 after completing the first rescue mission of task 1,2 and 3. It became available at 17:55 after necessary preparations.

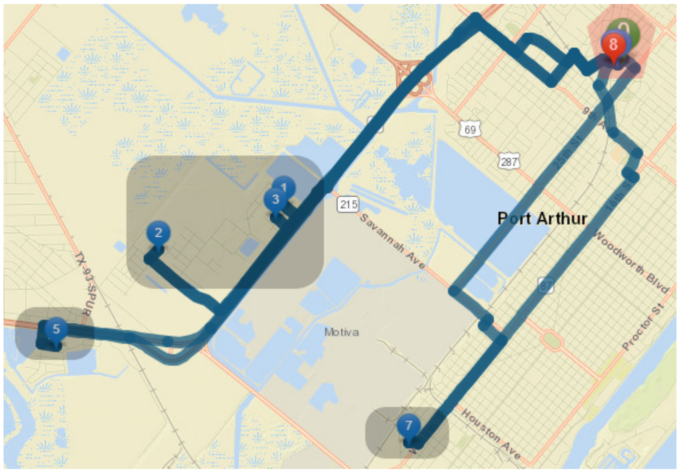


Figure 5. Route of Rescue Unit 1 by rescue order

The routes of the rescue missions assigned to rescue unit 1 presented in Table 11 are illustrated in Figure 5. The red pentagon shadow area denotes the rescue operation base. The black shadowed rectangular shapes represent the rescue mission. Location points 1, 2, 3, 5 and 7 denote the taskSeq 1,3, 2, 8 and 6, respectively. Pointers 1,2, and 3 are inscribed in the same box as those tasks were combined together and performed in a single mission. The rescue unit 1 will start from the base (0) and travel to point 1, 2 and 3 to rescue victims and complete the tasks 1,3 and 2 in the first rescue mission. It will return back to base which is denoted by blue pointer (4) below the red pointer indicating 8. The unit will again travel to location 5, return to the base (6) and complete the taskSeq 8. Finally, the location of the third rescue mission pointed by 7 and the missions will be completed by rescue unit 1 after reaching to the base (point 8).

We have also conducted the same experiment with 4 rescue units. The average waiting time and turnaround time reduced dramatically in this scenario. In the first experiment with 2 rescue units, the average waiting time and turnaround time is around 137 minutes and 189 minutes respectively. With 4 rescue units, waiting time and turnaround time came down to 49 minutes and 102 minutes. With the low number of rescue units, the tasks with low priority need to wait longer which increase the average waiting time. From Table 10 and Table 11, we can observe that taskSeq 5 arrived at 14:46 with a priority score of 1. Due to the very low priority, task 5 scheduled last at time 20:49 with a waiting time of 375 minutes. However, tasks with higher priority such as 1, 7 and 8 had to wait a fairly lower amount of time.

6. CONCLUSION AND FUTURE WORK

In this paper, we utilized social media (Twitter) for disaster management applications such as categorizing, identifying, and prioritizing users who need help and developed an algorithm for rescue scheduling. We introduced a novel approach for an effective rescue scheduling algorithm. First, we developed a tweet classifier using deep learning with attention layer and auxiliary features. The classifier labels every tweet into six different classes. Those labels allow us to identify the necessary information to assist the person/people in the tweet and estimate a priority score for that task. Second, we developed a multi-task hybrid scheduling algorithm and conducted the experiments using real disasters data for evaluating the efficiency of the algorithm. In the future, we would like to work on precise location determination and optimal estimation of the required time for a rescue mission. In addition, we are developing a fully-featured web application for deploying on the real-time disaster to evaluate the effectiveness of our work in disaster management.

ACKNOWLEDGEMENTS

This research has been partially supported by a grant from NSF CNS-1461914.

REFERENCES

[1] Mina Park, Yao Sun, and Margaret L McLaughlin. Social media propagation of content promoting risky health behavior. *Cyberpsychology, Behavior, and Social Networking*, 20(5):278–285, 2017.

[2] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):3, 2018.

[3] David E Alexander. Social media in disaster risk reduction and crisis management. *Science and engineering ethics*, 20(3):717–733, 2014.

[4] Zhou Yang, Long Hoang Nguyen, Joshua Stuve, Guofeng Cao, and Fang Jin. Harvey flooding rescue in social media. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 2177–2185. IEEE, 2017.

[5] Irina Shklovski, Moira Burke, Sara Kiesler, and Robert Kraut. Technology adoption and use in the aftermath of hurricane katrina in new orleans. *american Behavioral scientist*, 53(8):1228–1246, 2010.

[6] Drake Baer. As sandy became# sandy, emergency services got social. *Fast Company*, 9, 2012.

[7] Bruce R Lindsay. Social media and disasters: Current uses, future options, and policy considerations, 2011.

[8] Long Nguyen, Zhou Yang, Jiazhen Zhu, Jia Li, and Fang Jin. Coordinating disaster emergency response with heuristic reinforcement learning. *arXiv preprint arXiv:1811.05010*, 2018.

[9] Jie Yin, Sarvnaz Karimi, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. Using social media to enhance emergency situation awareness. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[10] Huiji Gao, Geoffrey Barbier, Rebecca Goolsby, and Daniel Zeng. Harnessing the crowdsourcing power of social media for disaster relief. Technical report, Arizona State Univ Tempe, 2011.

[11] Kate Starbird and Leysia Palen. *Pass it on?: Retweeting in mass emergency*. International Community on Information Systems for Crisis Response and . . . , 2010.

[12] Ory Okolloh. Ushahidi, or "testimony": Web 2.0 tools for crowdsourcing crisis information. *Participatory learning and action*, 59(1):65–70, 2009.

[13] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[14] Eui-Hong Sam Han, George Karypis, and Vipin Kumar. Text categorization using weight adjusted k-nearest neighbor classification. In *Pacific-asia conference on knowledge discovery and data mining*, pages 53–65. Springer, 2001.

[15] Mirella Lapata, Phil Blunsom, and Alexander Koller. Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 1, long papers. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, 2017.

[16] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[17] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. *arXiv preprint arXiv:1605.05894*, 2016.

[18] Dat Tien Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Rapid classification of crisis-related data on social networks using convolutional neural networks. *arXiv preprint arXiv:1608.03902*, 2016.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[20] Joseph Y-T Leung and Jennifer Whitehead. On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance evaluation*, 2(4):237–250, 1982.

[21] Wen-Hung Huang and Jian-Jia Chen. Self-suspension real-time tasks under fixed-relative-deadline fixed-priority scheduling. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*, pages 1078–1083. IEEE, 2016.

[22] Bin Hu, Fang Pan, and Lei Wang. A scheduling algorithm for medical emergency rescue aircraft trajectory based on hybrid estimation and intent inference. *Journal of Combinatorial Optimization*, 37(1):40–61, 2019.

[23] Felix Wex, Guido Schryen, Stefan Feuerriegel, and Dirk Neumann. Emergency response in natural disaster management: Allocation and scheduling of rescue units. *European Journal of Operational Research*, 235(3):697–708, 2014.

[24] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*, 2015.

[25] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[26] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[27] Akshi Kumar, Saurabh Raj Sangwan, Anshika Arora, Anand Nayyar, Mohamed Abdel-Basset, et al. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access*, 7:23319–23328, 2019.

[28] Xingyou Wang, Weijie Jiang, and Zhiyong Luo. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437, 2016.

[29] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[30] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 994–1009. ACM, 2015.

[31] Md Kabir, Sanjay Madria, et al. A deep learning approach for tweet classification and rescue scheduling for effective disaster management. *arXiv preprint arXiv:1908.01456*, 2019.

[32] ArcGIS. Arcgis for developers, mar 2019. URL `https://developers.arcgis.com/javascript/`.

II. STIMULATE: A SYSTEM FOR REAL-TIME INFORMATION ACQUISITION AND LEARNING FOR DISASTER MANAGEMENT

Md Yasin Kabir and Sanjay Madria
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri 65401
Email: mkabir@mst.edu and madrias@mst.edu

ABSTRACT

Real-time information sharing and propagation using social media such as Twitter has proven itself as a potential resource to improve situational awareness in a timely manner for disaster management. Traditional disaster management systems work well for analyzing static and historical information. However, they cannot process dynamic streams of data that are being generated in real-time. This paper presents STIMULATE - a System for Real-time Information Acquisition and Learning for Disaster Management that can (1) fetch and process tweets in real-time, (2) classify those tweets into FEMA defined categories for rescue priorities using pre-trained deep learning models and generate useful insights, (3) find FEMA defined stranded people for rescue missions of varying priorities, and (4) provide an interactive web interface for rescue management given the available resources. The STIMULATE prototype is primarily built using the Python Flask framework for web interaction. Additionally, it is deployed in the cloud environment using Hadoop and MongoDB for scalable storage, and on-demand computing for processing extensive social media data. The deep learning models in the STIMULATE prototype use Python Keras and the TensorFlow library. We use Bi-directional Long Short-Term Memory (BLSTM) and Convolutional Neural Network (CNN) for developing the tweet classifier. Further, we use the Python PyWSGI WebSocket server for rescue scheduling operations. We present a deep learning system trained on hurricane Harvey and Irma datasets only. The tweet classifier is evaluated using 15 different disaster datasets. Finally, we present the results of multiple simulations using synthetic data with different sizes to measure the performance and effectiveness of the tweets processor and rescue scheduling algorithm.

Keywords: Real-time system, Disaster management, Rescue scheduling, Deep learning, Social media;

1. INTRODUCTION

Real-time Information and insights play a vital role in disaster planning, response, and recovery. Effective communication can reduce the impact of a disaster and decrease the fatality rate. However, natural disasters frequently disrupt regular communication and damage or jam communication infrastructures[1] which leads to an outflow of information which is essentially required. Non-traditional communication systems like social media become more interactive and provide a continuous flow of event-based information. During hurricane Harvey more than 56,000 calls came into 911 within 15 hours overwhelming the emergency response system. A report on disaster "Sandy"[2] shows that people are using social media more frequently to communicate and seek help. Social media allows people to communicate promptly as people strive to contact friends and family, looking for information regarding transport, shelter, and food. During hurricanes Harvey and Irma, Twitter played a crucial role in the rescue, donations, and disaster recovery[3]. Hence, the huge flow of information over social media can be beneficial in regards to managing a natural disaster. However, while the use of social network seems appealing most of the applications which use social media are still lacking in features and fall short in usability [4].

During a disaster, institutional and volunteer rescue efforts save many lives. After the devastating hurricanes Harvey and Irma lots of people participated in the rescue mission as volunteers using social media[5]. However, such missions are typically not organized or structured. Moreover, individual volunteers lack resources, advanced equipment, and medical facilities. Besides, there is a high chance of missing rescue seeking tweets without a system for fetching and analyzing a huge stream of tweets in real-time. To address this issue, an automated system is crucial to fetch and process tweets in real-time, extract useful information and detect the tweets seeking help, and then arrange rescue missions given the resources. We conduct extensive research on how to create such an automated system to leverage machine learning and social media (Twitter) for effective disaster management. In this system paper, we discuss a novel disaster management system named STIMULATE which can facilitate real-time social media data collection, processing, and analysis along with a user friendly web interface for rescue management. The primary components of the proposed systems are:

- A multi-headed binary classifier developed using deep neural networks which classify the tweets into six different classes that are considered vulnerable populations according to FEMA[1].
- An easy-to-use web interface which allows filtering tweets by multiple keywords or specified geo-areas. Further the system extracts and estimates the location of classified tweets and determines the priority using classified labels and live weather APIs.
- An intuitive and interactive web interface for rescue management where along with institutional efforts, volunteers can also register and participate in rescue missions.

2. RELATED WORKS

Yang, Zhou, et al. [5] proposed a rescue scheduling algorithm recently based on data from hurricane Harvey. This algorithm connects the victims with scattered rescue volunteers. However, synchronized rescue efforts along with the government and organization might be more effective and faster. Jie et al. [6] proposed a system that uses data

[1] https://fema.gov/news-release/2019/05/17/fact-sheet-not-all-disaster-preparedness-plans-are-same

mining and machine learning mechanisms to extract the data generated by Twitter messages during crisis. Gao et al. [7] presented research showing how social media communication was used during the catastrophic Haiti earthquake. The authors also present a fact that the messages and photos shared on the micro-blogs by many individuals help the affected area to raise a huge fund. They adapted the method of crowd-sourcing for designing coordination protocols and mechanisms to create coordination between the organizations and their relief activities. Palen et al. [8] analyzed the extensive use of Twitter data in case of mass convergence or disaster situations such as the Southern California Wildfires.

There is a scarcity of automated disaster management systems. However, after several devastating incidents, a few applications are developed. Ushahidi [9] is such an example, where the recent incidents and an emergency situation can be tracked along with the geographical map. Irmamiami [2] is a live application using the Ushahidi's open-source code depicting the different activities in the Miami area related to hurricane Irma. In [10], the authors propose a distributed stream processing platform to support real-time analytics of social media data during a disaster. There are several applications such as TweetDeck [3], TwitInfo [11], and Twitcident [12] that provide functionalities for tracking, and analyzing tweets in real-time on various ongoing events. Although, these applications provide partial functionalities to collect and process real-time streaming data, they lack various crucial components for an automatic disaster management system which can detect help-seeking people from social media posts and coordinate rescue operations among institutional efforts, individual volunteers, and victims. In this work, we propose a system which can address those issues with an interactive and robust web application.

[2] https://irmamiami.ushahidi.io

[3] https://tweetdeck.twitter.com/

3. STIMULATE ARCHITECTURE

The STIMULATE Architecture primarily consists of three modules as shown in Figure. 1: A) Tweet Fetcher module, B) Tweet Processing module, and C) Rescue Scheduling module. In the following subsections we briefly describe the different modules of the proposed system. A comprehensive description and evaluation of each module and their algorithmic techniques are available in our previous research work [13].

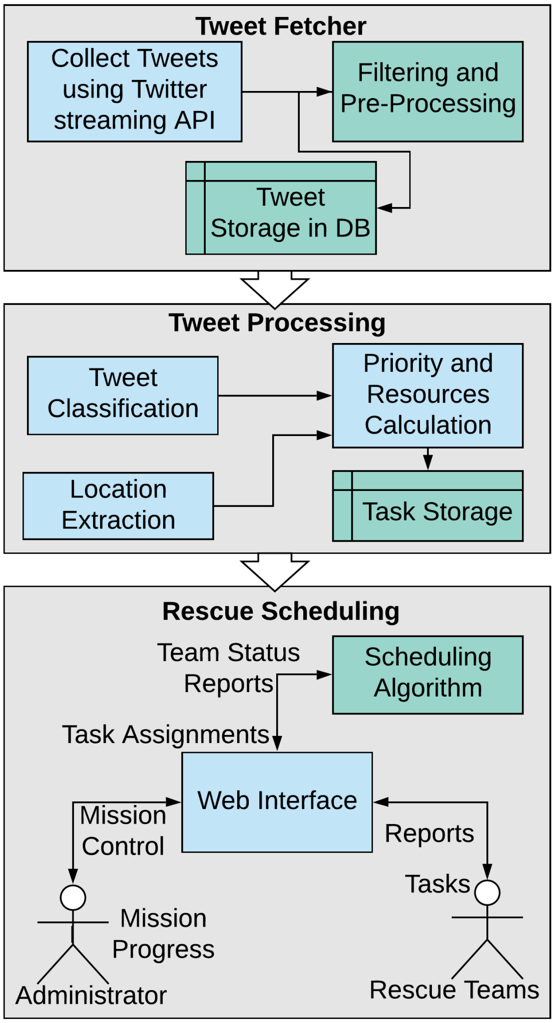


Figure 1. System architecture of the STIMULATE

3.1. TWEET FETCHER

The tweet fetching module obtains a stream of tweets using the Twitter Streaming API. Tweets from this stream can be filtered by keywords and location. Each tweet is then processed and streamlined for classification - emojis are replaced with category keywords, slang and jargon are replaced with more common wordings, and contractions are expanded. This module also extracts several auxiliary features from each tweet, including punctuation frequency, emoji category frequency, and percentage of capital letters. The tweet text and extracted features are then passed to the tweet processing module.

3.2. TWEET PROCESSING

The tweet processing module analyzes and classifies the incoming tweets in order to find stranded individuals and determine the task parameters for their rescue. This module has three main functionalities: tweet classification, location extraction, and priority determination. Tweet classification is done by a pre-trained neural net, and determines various parameters about a stranded individual's situation. After classification, the individual's location is determined. Finally, based on classification results and various situational factors, the rescue priority is determined. A task specification is created from this data and passed on to the rescue scheduling module.

3.2.1. Tweet Classification. During the tweet classification stage, tweets are determined to fall into one or more of six classes (as defined by FEMA): Rescue Needed, DECW (Diseased, Elderly, Children, and pregnant Women), Water Needed, Injured, Sick, and Flood. The Rescue Needed class identifies that a tweet includes a call for help. The remaining classes help determine priority and needed supplies: the DECW indicates vulnerable or at-risk individuals, the Water Needed class indicates a need for water, while the Injured and Sick classes indicate a need for medical supplies, and the Flood class indicates a common environmental hazard. These classes help determine rescue mission priority and needed supplies. Tweets are classified into these categories by a deep neural network that combines attention-based a Bi-directional Long Short-term Memory (BLSTM) Network with a Convolution mechanism. The BLSTM network is trained on understanding complex language based both on the tweet text and the auxiliary features extracted in the previous module. The attention mechanisms allow the network to identify words with the closest semantic relationships within the text. Finally, the convolution mechanism merges the word-vectors into a features map and generates the output as a set of binary values for each of the six classes. Table 1 represents the primary parameters of the model. A more detailed description of the developed classification model is available in [13].

Table 1. Hyperparameter values

Hyperparameter	**Value/Description**
BLSTM Layer	2 layers; 300 units (Forward and Backward)
Conv1D Layer	3 layers; 300 convolution filters
Dense Layer	3 layers with number of units 150, 75, and 1.
Drop-out rate	Word Embedding: 0.3; Dense layer: 0.2 each;
Activation function	ReLU and Sigmoid; Optimizer: Adam.
Epochs and batch	Epochs = 10 to 25; batch size = 128;

3.2.2. Location Extraction. Due to Twitter's privacy policy, most tweets do not contain location information in the meta-data; however, most rescue requests include an address or location of some kind, which is extracted using the Stanford Named Entity Recognizer. Twitter profile metadata can also contain hints to the location of the user. With the help of the Twitter metadata, user profile, and Google Maps API, the module estimates the location of the stranded individual.

3.2.3. Priority Determination. An effective rescue operation requires awareness of the higher urgency of some rescue requests - individuals in immediate danger must be rescued immediately. The priority of a rescue request is calculated based on that request's classification as well as additional situational factors. A vector of weights for the six classes is used to calculate the priority of a task based on the classes it falls into. Additionally a feature vector is used with weights for other considerable factors, such as the number of victims, environmental conditions, and weather forecasts. Weights relevant to the request are summed together to calculate the priority, which can range from a base priority of 1 to the highest priority of 10. Equation 1 represents the formula to estimate the priority for a rescue task.

$$f_p = \sum_{i=1}^{m} \alpha_i + \sum_{j=1}^{n} w_j \tag{1}$$

3.3. RESCUE SCHEDULING

The rescue scheduling module provides tools to effectively manage a rescue operation. The module consists of several components: the task scheduling algorithm determines the optimal order of rescue operations, the client console provides a web-based user interface for rescue teams to manage their tasks, and the administrator console provides a web-based user interface for monitoring mission progress. In addition to these components there are two server applications – a Flask-based Python HTTPS server for displaying the consoles, and a PyWSGI-based WebSocket server that handles communications between the scheduling module and rescue teams. Driver code written in Python connects these components together. The scheduling module reads in task specifications from a data file in real time while the tasks are being extracted by the other modules. The driver code translates task specifications into Task objects – collections of task-related data – and provides them to the scheduling algorithm.

3.3.1. Scheduling Algorithm. Selecting a scheduling algorithm to manage rescue operations can be difficult due to the dynamic nature of a rescue situation and the much greater cost of task “failure”. It is important that the algorithm is sensitive to the differing criticality of certain rescue tasks – an individual in immediate danger must be rescued immediately, whereas some individuals can wait to be rescued. From these considerations we determine that a priority-based scheduling algorithm would be most effective, as it exhibits the proper fairness for a rescue situation. To improve the efficiency of the rescue teams, we also add a multi-task feature to the algorithm which allows it to assign additional lower-priority tasks to a single rescue team if they can all be accomplished alongside the top priority task being scheduled. The result is the *Hybrid Multi-Task Priority Queue (HMTPQ)* algorithm depicted in Figure 2.

The scheduling module implements the HMTPQ algorithm to schedule the tasks it receives from the tweet processing module. The scheduling module creates a priority queue for the tasks it needs to schedule, keyed on task priority. The top priority task is selected and assigned to the first available rescue team which has the necessary supplies and capacity for the task. For a low priority task (priority less than or equal to seven), the algorithm continues searching for more tasks for the rescue team. It iterates through the queue looking for lower-priority tasks within a two-mile radius of the initial task and assigns them if the rescue team still has the supplies and capacity to perform them. Once the queue is exhausted or the rescue team reaches its limit, the scheduling algorithm orders the assigned tasks by priority and then instructs the WebSocket server to alert the rescue team of their assignment. The algorithm then returns to the next top task in the priority queue and searches for a rescue team to complete the task.

The scheduling algorithm works with Task objects. These are collections of task-relevant data: a unique task label, the priority value, arrival time, estimated task duration or burst time, task location, resource requirements, and number of people to rescue. The Task object additionally provides comparison operators to enable ordering tasks: a task with a higher priority is considered “greater” than a task with a lower priority; if both priorities are equal, a task with a lower arrival time (an older task) is considered greater than a task with a higher arrival time (a younger task); if priorities and arrival times are equal, a task with a lower burst time is considered greater than a task with a higher burst time.

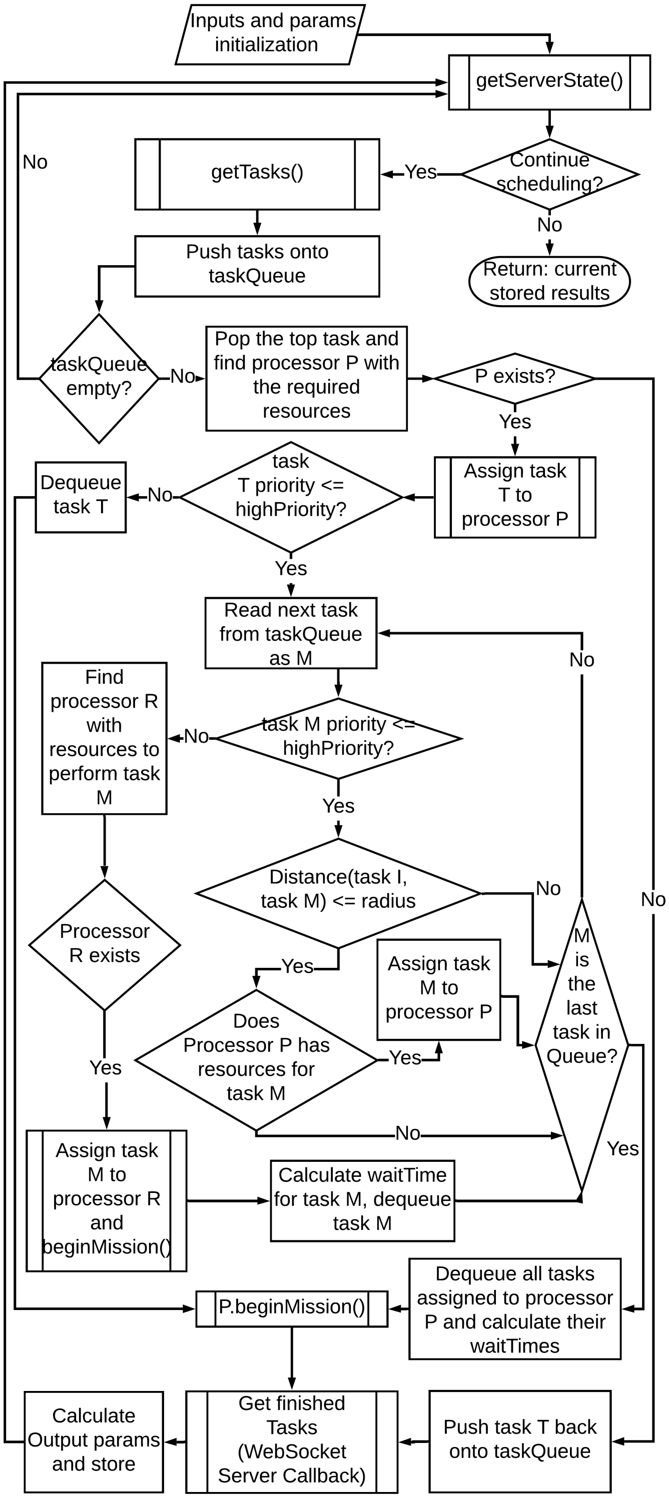

Inputs and params initialization
getServerState()
Continue scheduling?
Yes
No
getTasks()
Push tasks onto taskQueue
Return: current stored results
No
taskQueue empty?
No
Pop the top task and find processor P with the required resources
P exists?
Yes
No
Assign task T to processor P
task T priority <= highPriority?
No
Dequeue task T
Yes
Read next task from taskQueue as M
No
task M priority <= highPriority?
No
Find processor R with resources to perform task M
Yes
Distance(task I, task M) <= radius
No
Yes
Assign task M to processor P
Processor R exists
Does Processor P has resources for task M
Yes
M is the last task in Queue?
Yes
No
Yes
Assign task M to processor R and beginMission()
Calculate waitTime for task M, dequeue task M
Dequeue all tasks assigned to processor P and calculate their waitTimes
P.beginMission()
Calculate Output params and store
Get finished Tasks (WebSocket Server Callback)
Push task T back onto taskQueue

Figure 2. Scheduling algorithm

Because the scheduling algorithm must access and remove elements from the middle of a priority queue which is not supported in most implementations, a custom priority queue is implemented in Python for this algorithm. The custom implementation is built as a binary maxheap and includes the standard priority queue functions pushing an item on to the queue, popping an item off the top of the queue, and enforcing the heap property. However, our implementation allows elements from any position in the queue to be read or removed. This enables the scheduling module to assign multiple tasks to a rescue team regardless of their priority.

3.3.2. Servers. The Flask and WebSocket servers are an important part of the scheduling module because they enable the remote human elements – rescue teams and administrators – to interact with the module. Communication between the scheduling module and the clients is handled by the WebSocket server. The Flask server provides a web-based user interface - which wraps the client-side socket communication component – for rescue teams and administrators. Both servers are implemented as one Python module, operating on different ports of the same server hardware.

The WebSocket server forms the back-end for WebSocket communication. It listens for connecting sockets and performs an authentication and handshake step, ensuring the validity of the connecting socket and determining its role (rescue team or administrator). The server implements a lightweight protocol on top of the WebSocket protocol for its client-side messages. The protocol supports sending message codes, as well as ordered and un-ordered collections of atomic values.

The WebSocket server provides callback functions for answering several types of client messages, each with their own specific parameter requirements. These callbacks process reports, updates, and commands from the human operators of the module and modify various system states and variables, which are made visible to the scheduling algorithm through the driver code. In this way the WebSocket server delivers real-time updates from the rescue effort directly to the scheduling module. The Flask server enables a web-based user interface served over HTTPS. This interface wraps the client-side of WebSocket communication, providing various buttons that trigger WebSocket messages to the server, as well as JavaScript listeners for messages from the server. The web interface enables the users to interact with the scheduling module in a human-friendly way.

3.3.3. Resource Management. The scheduling module is responsible for managing resources dynamically in real time. Doing so precisely is important so that the scheduling algorithm can accurately tell which teams have the necessary resources for the tasks.

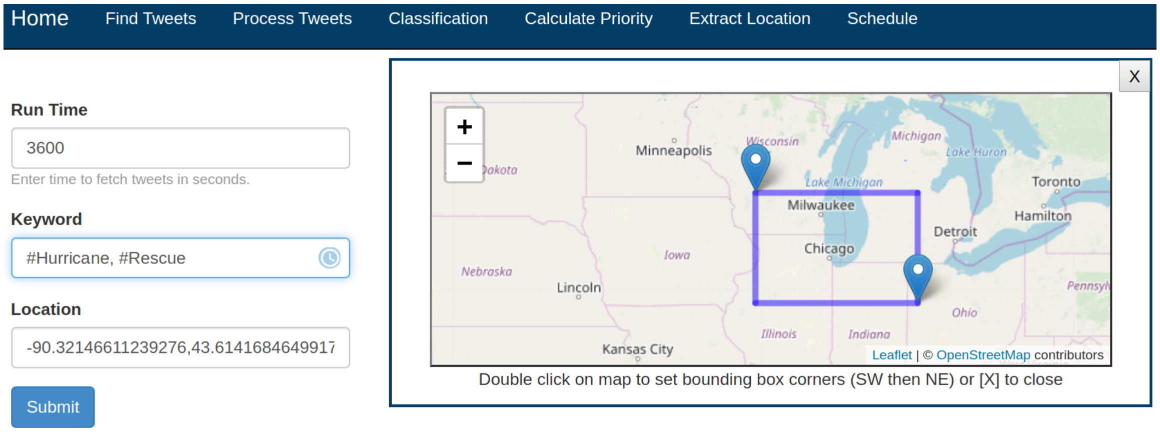
Automatic resource management is handled by the scheduling algorithm, which reduces the resources of a registered rescue team when assigning a task - these are considered reserved resources for that task. This reduction is also reflected in the rescue team's web interface. The manual resource management happens primarily in the client console. Rescue teams are able to add new resource and manage resource amounts through the inventory table. All updates to the table are immediately reported to the server and algorithm.

3.3.4. Data Preservation. This module generates two log files for data preservation. One is a sequential log of server output, and the other contains the latest task schedule. These files are updated in real-time and persist on the system in the event of an unexpected shutdown.

4. STIMULATE PROTOTYPE DEMONSTRATION

4.1. REAL-TIME DATA COLLECTION AND PROCESSING

Figure 3 depicts the web-based interface for the tweet fetcher module. This interface allows the user to specify a finite time limit for the fetching, and keywords and a geolocation to filter by. The run time can be set as infinite and the module will collect tweets, extract and clean tweet data, and store it in data files indefinitely. Further, it will pass this data to the tweets processor which will analyze the tweets, extract useful information and auxiliary features, and pass it to the tweet classifier for detecting the desired tweets and classify those tweets into different categories. After classification the module passes the labeled tweets to the location estimation module and further to the priority detector. The location estimator extracts or estimates the location of the victim(s). If it is not able to estimate a location it will notify the administrator. The tweets processor also generates several useful insights about the ongoing disaster such as distribution of the fetched tweets over the time, plots of the tweets according to the location on a map, and heat-maps.



Run Time

3600

Enter time to fetch tweets in seconds.

Keyword

#Hurricane, #Rescue

Location

-90.32146611239276,43.6141684649917

Double click on map to set bounding box corners (SW then NE) or [X] to close

Figure 3. System prototype UI for fetching tweets

Figure 4 represents a plotted map of fetched tweets during hurricane Irma around Florida where the red dots denote the location of the individuals.

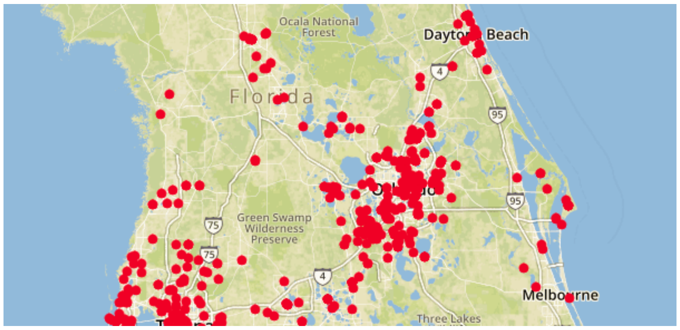
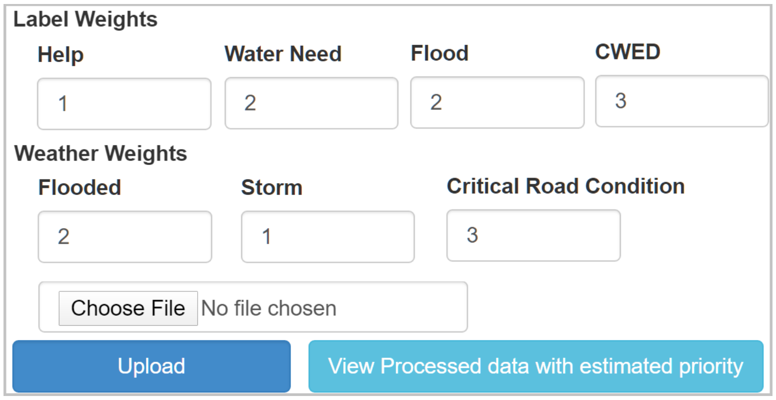


Figure 4. Tweets geo-location of hurricane Irma around Florida

4.2. PRIORITY DETERMINATION

The simplified priority determination module presented in Figure 5 allows the user to specify the weights for the different labels and environmental conditions such as flood, storm, and road condition. However, the module also seamlessly sets the weights of the labels and environmental variables from the predefined sets of rules and values. It uses the information from Open Weather API and determines the critically of the various weather situations. Further, using the priority determination equation it calculates the priority of each task and sends the information to the rescue scheduling module for final rescue coordination.



Label Weights

Help	Water Need	Flood	CWED
1	2	2	3

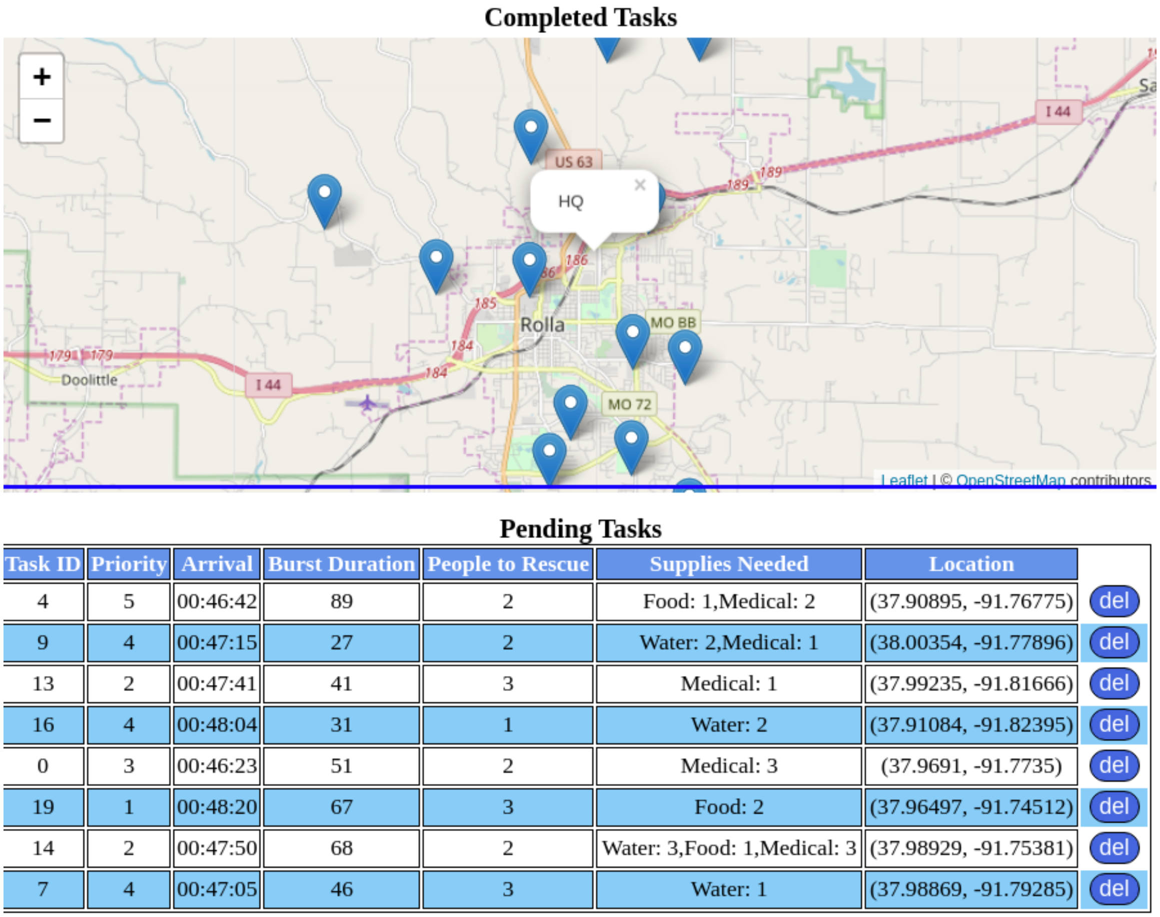
Weather Weights

Flooded	Storm	Critical Road Condition
2	1	3

Figure 5. Priority determination prototype demonstration

4.3. RESCUE SCHEDULING

4.3.1. Administrator Interface. The scheduling module allows for a human administrator to monitor rescue mission progress and issue several commands to the server through a web-based UI, demonstrated in Figure 6. The main view comprises of four panels: Rescue units list, Map panel, Completed tasks panel, and Pending tasks panel, populated automatically from server messages. The rescue units panel lists all registered rescue teams and their status, assigned tasks (if any), vehicle type, and supplies. The map panel includes two tabs: the completed tasks map shows the locations of all completed tasks, whereas the in-progress map shows all ongoing tasks and routes. The in-progress map can also be double clicked to manually create a rescue task at that location.Both maps are implemented using the Leaflet API.



Pending Tasks

Task ID	Priority	Arrival	Burst Duration	People to Rescue	Supplies Needed	Location	
4	5	00:46:42	89	2	Food: 1,Medical: 2	(37.90895, -91.76775)	del
9	4	00:47:15	27	2	Water: 2,Medical: 1	(38.00354, -91.77896)	del
13	2	00:47:41	41	3	Medical: 1	(37.99235, -91.81666)	del
16	4	00:48:04	31	1	Water: 2	(37.91084, -91.82395)	del
0	3	00:46:23	51	2	Medical: 3	(37.9691, -91.7735)	del
19	1	00:48:20	67	3	Food: 2	(37.96497, -91.74512)	del
14	2	00:47:50	68	2	Water: 3,Food: 1,Medical: 3	(37.98929, -91.75381)	del
7	4	00:47:05	46	3	Water: 1	(37.98869, -91.79285)	del

Figure 6. Admin interface displaying completed and pending tasks.

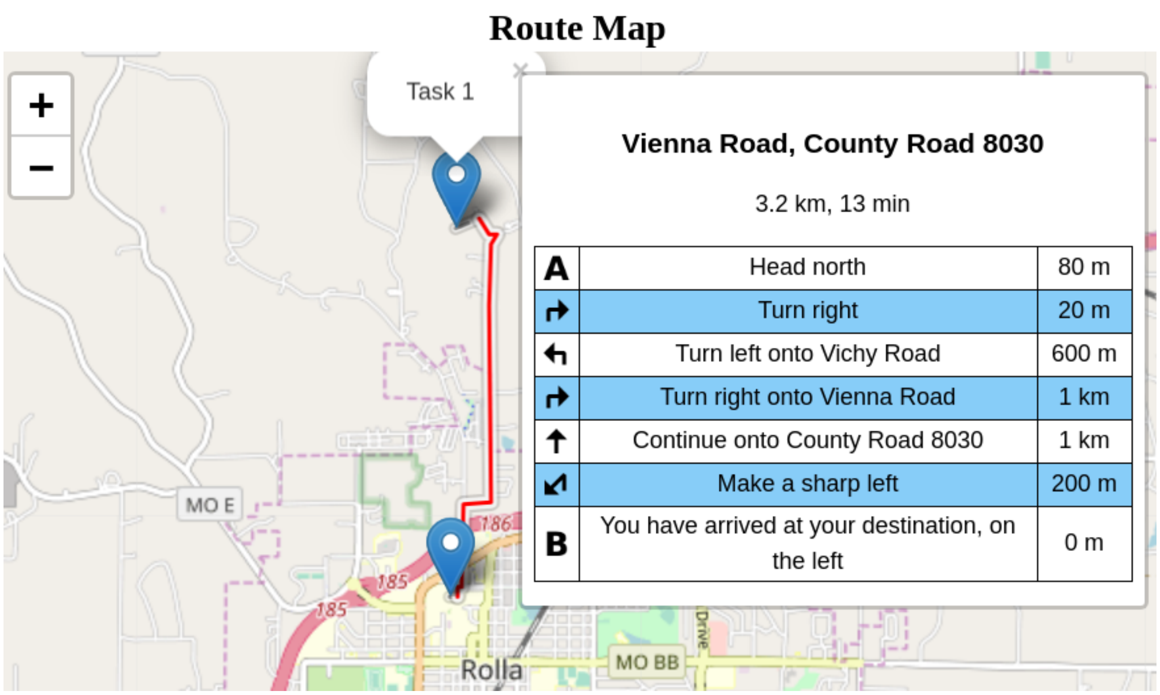
The completed tasks list shows the current task sequence, the rescue team responsible for each task, as well as timing information: arrival times, start times, wait times, finish times, and turnaround times. This panel also shows average wait and turnaround times. The locations of these completed tasks can be seen in the completed tab of the map panel. The pending tasks panel shows all tasks still waiting to be assigned. Each task in the panel includes its priority, arrival time, burst time, people to rescue, supplies needed, and location.

4.3.2. Rescue Team (Client) Interface. The scheduling module allows rescue teams to join and leave the rescue effort and to manage their tasks dynamically through the client console. The client console allows teams to report on task status, update their supplies inventory, and plan their rescue routes. Reports are sent to the WebSocket server which manages the list of registered rescue teams (providing it to the scheduling algorithm and displaying it for the administrator) and responds to task status by recording task finish times for successful missions and pushing failed tasks back on the queue.

Rescue teams operate according to a cycle of three states: Available state, Busy state, and Restocking state. The available state is active before an assignment and indicates the team is available for new tasks, the busy state is active during an assignment and indicates the team is in progress on a mission, and the restocking state is active after an assignment and indicates that the team is resupplying after a mission. Tasks can only be assigned to teams in the available state.

Rescue teams must fill out a registration form with details on their vehicle and inventory, before they are shown the main view, which comprises of four components: the tasks checklist, the supplies inventory, and the route map. The tasks checklist shows all tasks currently assigned to the team, in the order in which they should be completed - high priority tasks at the top. Each task has a checkbox next to it which can be checked to indicate successful completion of the task, or left unchecked to indicate the task is not completed. The checklist will be empty if no tasks are currently assigned to the team or the team has indicated that it needs more time to prepare for tasks. The check-in button will become available when the server assigns a mission, and can be clicked to report checked tasks as successful and unchecked tasks as failed.

The route map depicted in Figure 7 is populated with a suggested route for visiting each task location in order. The map is generated using the Leaflet API. Routes are generated using the Open Source Routing Machine (OSRM) API. The map is interactive - new destinations can be added along the route and existing destinations can be moved around in response to sudden road condition changes.



Route Map

Vienna Road, County Road 8030

3.2 km, 13 min

A	Head north	80 m
↱	Turn right	20 m
↰	Turn left onto Vichy Road	600 m
↱	Turn right onto Vienna Road	1 km
↑	Continue onto County Road 8030	1 km
↙	Make a sharp left	200 m
B	You have arrived at your destination, on the left	0 m

Figure 7. Client interface of assigned task

The supplies inventory is used to keep track of the rescue team’s resources - food, water, and medical supplies. Updates to the supplies inventory are automatically reported to the server, which provides updated inventories to the scheduling algorithm to determine which teams are capable of which tasks. Each entry in the supplies inventory table can be incremented, decremented, removed and new entries can be added. Attempting to add a "new" entry of an existing resource will increase the existing amount of that resource.

During a typical rescue mission, the team will either start out in the available state or activate it by clicking the ready button. Once the scheduling algorithm determines a list of tasks for the team to complete, it will alert the team and activate the busy state. The team will perform the tasks, return to headquarters, and then report to the scheduling module using the check-in button, which will also activate the re-stocking state. After resupplying and preparing for the next assignment, the team will click the ready button to activate the available state. The team can log out of the system at any time, which will mark all active tasks as failed.

4.4. EXPERIMENTAL EVALUATION

We perform rigorous experimental evaluation of the performance and stability of each of the modules. We use a machine with Intel® Core™ i9-9900K CPU, Nvidia GeForce RTX 2080Ti, 64GB 2600MHz DDR4 RAM, 1TB of SSD, and Ubuntu 18.04.3 LTS OS.

‘

Table 2. Classifier evaluation of the deep learning model

Disaster Data	**Model Accuracy (%)**
Hurricane Harvey and Irma	93.7
Nepal Earthquake	87.5
California Earthquake	83.6
Typhoon Hagupit	88.3
Cyclone PAM	92.6
CrisisLex	93.6

Table 2 shows the evaluation results of the developed deep neural network model. We train the model on the hurricane Harvey and Irma data that we had labeled manually. However, we evaluate the model for different disaster data-sets from the CrisisNLP [14] and CrisisLex [15] repositories to observe the robustness of the model. The accuracy results show that the developed model is performing well across multiple data-sets. A more detailed evaluation is available in [13].

Table 3 presents the performance evaluation of the modules for different data sizes. The tweet processor was able to process 500,000 tweets in around 18 minutes while it extracted 64 different attributes from each tweet. The other modules are also able to complete their tasks for different sample sizes in a fair amount of time. We were not able to measure the performance of priority calculator and location estimator module for 100,000 and

500,000 samples due to API credit limitations. These modules depend on the Open Weather and Google Maps APIs, which impose strict limits on the number of API calls. However in a real-life disaster situation our system would process the fetched data simultaneously in every few seconds and only perform priority calculation and location estimates on tweets classified as requests for help, so it is highly unlikely that it will exceed the API call limits.

Table 3. Performance evaluation of the STIMULATE modules

Module Name	**Sample Sizes and Time Taken(sec)**			
	1K	**10K**	**100K**	**500K**
Tweets Processor	7.29	81.45	447.97	1075.14
Tweets Classifier	86.27	945.37	4720.33	9541.66
Priority Calculator	7.28	73.68	NA	NA
Location Estimator	75.35	753.12	NA	NA

In Table 4, we present the performance of the rescue scheduler. We use two scripts to simulate a real-time rescue situation: one to generate tasks and one to spawn automated rescue units. The rescue units each have an associated WebSocket which they use to connect to the module and receive and complete tasks. We use this simulation to observe how our system performs when scheduling large numbers of tasks with various requirements among a pool of "real" rescue units.

Table 4. Performance of the Scheduling Algorithm

# Processors	**# processes and time taken (sec)**			
	100	**200**	**500**	**1000**
10	8.650737	22.431122	39.931425	76.669116
20	7.821158	12.885503	24.509293	67.108326
50	7.901736	9.624312	16.267383	76.317417

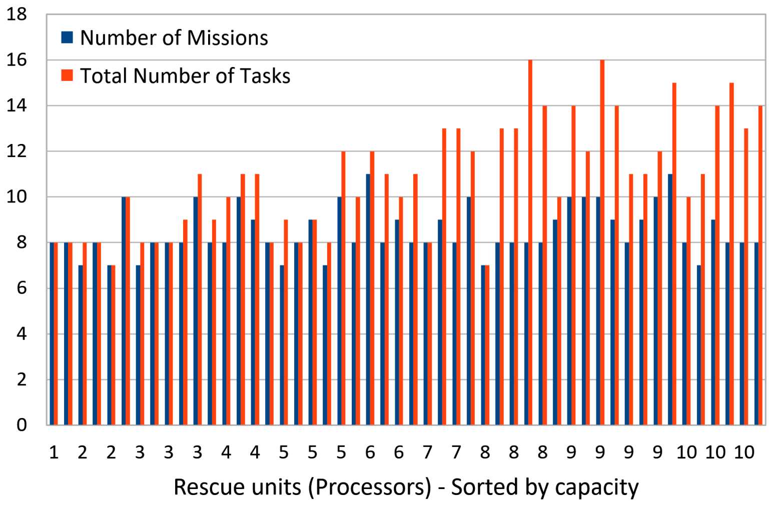


Figure 8. 500 Tasks distribution among 50 rescue units

Figure 8 depicts the allocation of the rescue missions for 500 tasks among 50 rescue units. In the figure, the red bar represents the number of tasks assigned to a processor, and blue bar represents the number of missions performed by a processor where a mission can consist of one or more tasks simultaneously assigned to the processor.

5. CONCLUSION

In this paper, we describe the architecture and demonstrate a working prototype of STIMULATE: a System for real-time Information Acquisition and Learning for Disaster Management. The system can collect, process, classify the tweets in real-time from streaming data and coordinate rescue operations among victims, institutional rescue efforts and individual volunteers with resource management. The simulation and performance evaluation of the system represents a stable and robust design. The developed prototype shows promise and can process datasets of considerable sizes quickly in real-time. Although the system has a functional prototype, there are still additional features which could be useful to develop for a full disaster management system, such as automatic combination and analysis of multiple sources of data. In the future, we will also implement a central database system in the cloud to store the historical data of disaster events so that we can produce useful insights for future disaster management and train our system on this data for improved accuracy.

ACKNOWLEDGEMENTS

This research has been partially supported by a grant from NSF CNS-1461914.

REFERENCES

[1] Irina Shklovski, Moira Burke, Sara Kiesler, and Robert Kraut. Technology adoption and use in the aftermath of hurricane katrina in new orleans. *American Behavioral Scientist*, 53(8):1228–1246, 2010.

[2] Drake Baer. As sandy became# sandy, emergency services got social. *Fast Company*, 9, 2012.

[3] Marcus Gilmer. During harvey, social media rose to the challenge as a force for good. *Mashable, August*, 29, 2017.

[4] Bruce R Lindsay. Social media and disasters: Current uses, future options, and policy considerations, 2011.

[5] Zhou Yang, Long Hoang Nguyen, Joshua Stuve, Guofeng Cao, and Fang Jin. Harvey flooding rescue in social media. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2177–2185. IEEE, 2017.

[6] Jie Yin, Sarvnaz Karimi, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. Using social media to enhance emergency situation awareness. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[7] Huiji Gao, Geoffrey Barbier, and Rebecca Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14, 2011.

[8] Kate Starbird and Leysia Palen. Pass it on?: Retweeting in mass emergencies. In *The 7th International Information Systems for Crisis Response and Management Conference, Seattle, WA, USA*, 2010.

[9] E Hirata, MA Giannotti, APC Larocca, and JA Quintanilha. Flooding and inundation collaborative mapping–use of the crowdmap/ushahidi platform in the city of sao paulo, brazil. *Journal of Flood Risk Management*, 11:S98–S109, 2018.

[10] Daniel Wladdimiro, Pablo Gonzalez-Cantergiani, Nicolas Hidalgo, and Erika Rosas. Disaster management platform to support real-time analytics. In *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–8. IEEE, 2016.

[11] Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 227–236, 2011.

[12] Fabian Abel, Claudia Hauff, Geert-Jan Houben, Richard Stronkman, and Ke Tao. Twitcident: fighting fire with information from social web streams. In *Proceedings of the 21st International Conference on World Wide Web*, pages 305–308, 2012.

[13] Md Yasin Kabir and Sanjay Madria. A deep learning approach for tweet classification and rescue scheduling for effective disaster management. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 269–278, 2019.

[14] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. *arXiv preprint arXiv:1605.05894*, 2016.

[15] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 994–1009, 2015.

III. EMOCOV: MACHINE LEARNING FOR EMOTION DETECTION, ANALYSIS AND VISUALIZATION USING COVID-19 TWEETS

Md Yasin Kabir and Sanjay Madria
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri 65401
Email: mkabir@mst.edu and madrias@mst.edu

ABSTRACT

The adversarial impact of the Covid-19 pandemic has created a health crisis globally all over the world. This unprecedented crisis forced people to lockdown and changed almost every aspect of the regular activities of the people. Thus, the pandemic is also impacting everyone physically, mentally, and economically, and it, therefore, is paramount to analyze and understand emotional responses during the crisis affecting mental health. Negative emotional responses at fine-grained labels like anger and fear during the crisis might also lead to irreversible socio-economic damages. In this work, we develop a neural network model and train it using manually labeled data to detect various emotions at fine-grained labels in the Covid-19 tweets automatically. We present a manually labeled tweets dataset on COVID-19 emotional responses along with regular tweets data. We created a custom Q&A roBERTa model to extract phrases from the tweets that are primarily responsible for the corresponding emotions. None of the existing datasets and work currently provide the selected words or phrases denoting the reason for the corresponding emotions. Our classification model outperforms other systems and achieves a Jaccard score of 0.6475 with an accuracy of 0.8951. The custom RoBERTa Q&A model outperforms other models by achieving a Jaccard score of 0.7865. Further, we present a historical emotion analysis using COVID-19 tweets over the USA including each state level analysis.

Keywords: COVID-19 data, Coronavirus, Twitter Data, Data analytics,Topics tracker, Emotion analysis, Machine learning.

1. INTRODUCTION

Every country is taking preventive measurements to fight against the COVID-19 pandemic. By the end of 2020, there were more than 83 million confirmed cases of novel coronavirus globally, and about 20 million people are infected[4] in the USA alone. The number of total fatal cases exceeded 1.8 million globally in 2020. The number of infected people and fatality keeps rising every day. Social distancing or stay-at-home became the most widely used directive all over the world. Social distancing is impacting public events, business activities, the educational domain, and almost every other activity associated with human life. People are losing their jobs and earning sources and thus, the stress level is rising at both the personal and community levels. The emotional responses became overwhelming and inconsistent as people are facing an unprecedented challenge. The studies of behavioral economics show that emotions can deeply affect individual behavior and decision-making.

Social networks have the hidden potential to reveal valuable insights on human emotions at the personal and community level. The monitoring of emotions at fine-grained labels could be valuable during and after the COVID-19 pandemic as the reactions of the people are changing every moment during this unpredictable time. The exploration of tweets to track emotions might play a significant role to understand people's behaviors and responses during the COVID-19 pandemic. The recent works [1, 2, 3, 4] show that Twitter data and human emotions analysis can be highly useful and it is not limited to only predict crimes, stock market, election polarity, and managing disasters. Therefore, it is paramount to analyze the social media data to understand the human behavior and reaction in the ongoing pandemic. To find out useful insights from the public reactions and shared posts in social media, and to model the public emotions, we have started collecting tweets from 5th March

[4] https://mykabir.github.io/coronavis/index.html

2020. We have collected and processed over 600 million tweets related to Coronavirus (focused on the USA only) which is more than 4.5 terabytes in raw data. We developed a web application that processes the collected data in real-time and produces interactive graphs and charts. The website is accessible publicly and enables anyone to observe the sentiments, topic trends, and user mobility with interactive visualizations including maps, time charts, and word clouds. Detailed information about the website and visualizations is available in [5].

There is a wide range of research works available where sentiments are explored using different techniques. Sentiments analysis [6, 7, 8, 9] became a popular field of natural language processing. In most of the sentiment analysis work, sentiments are explored considering high-level emotion categories such as positive, neutral, and negative. Several works also considered sentiment as a form of feeling using numerical scores such as 1 to 5 defining very bad to very good or something like that. However, to understand the emotional response of the people and correlate that with the socio-economic situation, we need fine-grained labels of emotion. For example, labeling the emotions like sadness, worry, or angry as negative sentiment only might not enable us to understand the proper reaction of a person as all three of those emotions may lead to different behaviors and decisions. Furthermore, while detecting and labeling the emotions into different categories is highly useful, it is also necessary to understand the reasoning behind an emotion. People might be angry or sad for different causes, and treating all of those causes similarly might not be ideal. To understand the reasoning behind an emotion, it is necessary to label a few words or a phrase from a text which will enable us to understand the emotions better and use them appropriately.

However, there is a lack of available labeled emotion data. In our research, we were able to find two available tweet emotion datasets. One of those datasets [10] has a total of 14,827 annotated tweets in 11 emotion categories (e.g. anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust) comprises with English, Arabic, and Spanish tweets. However, this dataset does not contain any COVID-19 related tweets.

The other dataset [11] which is annotated using COVID-19 tweets contains 10K English tweets and 10K of Arabic tweets in 10 different categories (e.g. optimistic, thankful, empathetic, pessimistic, anxious, sad, annoyed, denial, official report, and joking). We found that this dataset is useful for developing machine learning models to automatically detect and classify the tweet's emotions. However, the 10K labeled English tweets for 10 different categories are fairly low for creating an effective machine learning model. Moreover, none of those datasets provide the selected words or a phrase denoting the reason for the corresponding emotions.

Due to the lack of available datasets, most of the research works on COVID-19 sentiments such as [12, 13, 5] are mostly limited to the positive, neutral, and negative sentiments or researcher rely on some available API or lexicon-based tools that provides emotion categories without understanding the proper context which essentially is not appropriate for fine-grained emotion analysis tasks. There is also a lack of available machine learning models to automatically classify the emotion in the tweets using the context. To resolve those problems, we started annotated COVID-19 English tweets manually to 10 different emotion categories (e.g. neutral, optimistic, happy, sad, surprise, fear, anger, denial, joking, pessimistic) as well as also select the words or phrases that are mostly responsible for the selected emotion label. The phrase selection makes our dataset unique as there is no other such data available on COVID-19 tweet emotion to the best of our knowledge. Our annotated dataset can be used with the conjunction of the available dataset by [11] for the similar emotion labels to classify the emotion of the tweets and train a better machine learning model. In this work, we not only presented our dataset but also develop and train machine learning models to detect the emotion of the tweets and extract phrase which is mostly responsible for the detected emotion. We explore and created custom pipelines for the classification and phrase extraction tasks and perform a comparative study of the model performance. The primary contributions of this work are:

- A multi headed binary classifier using deep learning to automatically classify the COVID-19 tweets into above specified 10 emotions. The classifier determine the high-level relationships among the labels, and extract a contextual representation of the tweets to detect different emotions. The developed classification model achieves a Jaccard score of 0.6475 with an accuracy of 0.8951 outperforming other systems.

- A custom Q&A roBERTa model to extract the phrase that is mostly responsible for a particular emotion on a tweet. The model predicts the positions of the start and end tokens from a given text that represent the specified emotion. The proposed model achieves a score of 0.7865 in Jaccard metrics.

- Manually labeled (by three annotators) 10,000 tweets into 10 different emotions (e.g. neutral, optimistic, happy, sad, surprise, fear, anger, denial, joking, pessimistic). Along with the labels, we also selected the phrase that might be responsible for the respective emotion.

- An experimental historical emotion analysis on COVID-19 tweets using the developed classification model.

2. RELATED WORKS

Throughout the recent years social media has seen a tremendous increase in its use during times of crisis. Many researchers from all around the globe are creating COVID-19 datasets using Twitter APIs [14, 15] Putting together millions of tweets composed of largely English tweets related to keywords like: covid, corona, pandemic, and quarantine similar to those used in our research for the initial collection of tweets. Researchers are investigating methods of promoting healthy social media use during times of pandemic similar to the COVID-19 outbreak. With the growing amount of open data available relating to the public opinion from platforms like Twitter, Facebook, Instagram, Snapchat, Tumblr, LinkedIn, Youtube, Twitch, and Reddit [16, 17] as grown exponentially. Researchers are looking into different methods with the hope of developing an effective method of utilizing all the public data available through Twitter. Some suggest that the possibility of reaching an accuracy of sentiment classification is between 60 - 80 percent [9, 18].

Twitter being especially popular for anomaly detection, response and communication monitoring during crisis (disease outbreaks [19, 20], hurricanes [21, 22, 23], floods [24], terrorist bombing [25], misinformation propagation [26, 27] and others [28, 29]). Lisa et al. [30] and Ramez et al. [31] presented their works on misinformation propagation and quantification during COVID-19 using twitter. The authors in [31] conclude, there is an alarming rate of medical misinformation and non-credible content sharing on Twitter throughout the pandemic. It is very crucial to quantify the misinformation on social media and take the necessary action to prevent unnecessary anxiety and medically harmful methods to fight against COVID-19.

Catherine et al. [32] are exploring the possibility of illustrating topics such as spreading of corona case, healthcare workers and personal protection equipment (PPE) and seventeen others using a pattern matching and topic modeling system with Latent Dirichlet Allocation (LDA). The authors are investigating the use of five methods of analysis on features like key terms and features, information dissemination and propagation and network behavior during COVID-19 pandemic. These produced a model that could detect high level topic trends in news briefings over time. Alaa et al. [33] also performed topic modeling using word frequencies and Latent Dirichlet Allocation (LDA) with the aim to identify the primary topics shared in the tweets related to the COVID-19. Choudhury et al. [34] developed a dataset of classified tweets for a more refined set of emotions. Using a hashtag word classification system the authors were able to classify millions of tweets quickly. An example of this would be the translation of the word smile into the class of joviality.

Although there are many works available on tweet classification and phrase extraction we found only a few attempts to classify the tweets emotion during the COVID-19 pandemic using context-based machine learning models as there is a lack of available datasets. Most of the traditional tweet emotional classification works [10, 35, 36] treat the problem as a text classification problem and rely on a large amount of labeled data and focus mostly on effective feature engineering. Baziotis et al. [37] and Meisheri et al. [38] who hold the first and second place of the multi-label emotion classification task of SemEval-2018 Task1, developed classifiers using a bidirectional LSTM with an attention mechanism. Using two different trained models: regularized linear regression and logistic regression classifier chain, Park et al. [39] try to classify the emotions for the same problem discussed above. The authors captured the correlation of emotion labels using logistic regression classifiers. However, none of those works perform emotion classification on a crisis datasets which might represents a wide verify of emotions with unbalance labeled data. Yang et al. [11] introduce a COVID-19 dataset and implemented XLNet, AraBert, and ERNIE for classifying the emotion in English, Arabic, and Chinese language text respectively which is the only available emotion classification work on the COVID-19 tweets or text. For phrase extraction there are several transformation based models [40, 41] available from different research works. However, to our knowledge there is no available phrase extraction work on tweets emotion.

While there are ongoing research works for emotion detection and classification using the tweets there is a lack of publicly available datasets. Moreover, in most of those works, researchers are trying to label and detect emotion categories only for the tweets. However, the phrase that is responsible for a particular emotion in a tweet might help us understand the tweets better and can allow us to dive deep into data mining on emotional response. There is also a lack of available machine learning model that is developed particularly for automatic emotion detection of the COVID-19 tweets. In this work, we present the EMOCOV dataset that provides emotion category labels along with the phrase responsible for that emotion. We also propose two different machine learning models:

one is for emotion classification using deep learning approach with attention mechanism and auxiliary features input, and another one for extracting the responsible phrase for that emotion using a custom Q&A roBERTa head.

3. DATA COLLECTION, ANNOTATION AND DESCRIPTION

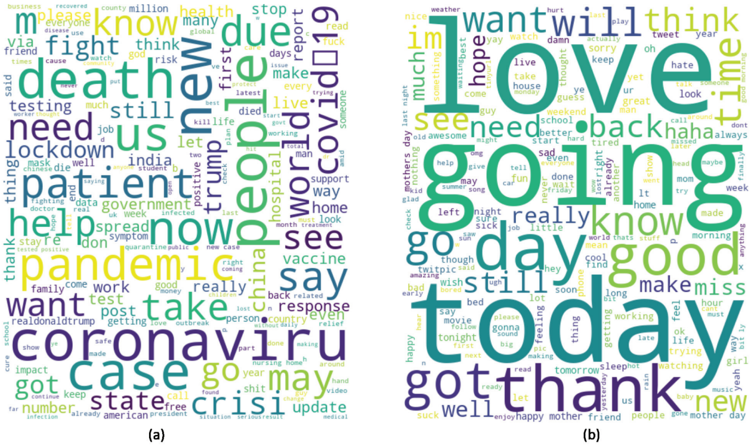


Figure 1. Word clouds from: (a) COVID-19 tweets, (b) Non-COVID tweets

At the early stage of our research, we have performed data analysis to observe and understand the differences between the available Twitter datasets for sentiment analysis and COVID-19 tweets. We observed that due to the ever-evolving nature of the tweets' linguistics and the newly allowed length of the tweet text (280 vs previous 140) there are noticeable contrasts between the available datasets and recent tweets. Moreover, during the ongoing pandemic, there is a frequent change in the events, guidelines, restrictions, news which creates a roller-coaster ride of emotions. Figure 1 represents the word clouds created using the tweet texts from the ongoing COVID-19 dataset and using a combined dataset created from the Crowdflower sentiment dataset and SemEval-2018 dataset. We randomly select 5000 data points from each category for generating word clouds. Figure 1(a) depicts the word cloud for COVID tweets, and Figure 1(b) represents the word cloud for the combined dataset of non-COVID tweets. From the figures, we can observe a good variation among the frequent words in the datasets. While general tweets contain usual words (e.g., love, going, today, thank) in the texts, COVID tweets are dominated by the words specifically related to the ongoing pandemic (e.g., death, patient, lockdown, death). We can also observe that only a few words in the non-COVID dataset are very frequent while the frequency of the top words in the COVID-tweets is much closer which is represents by the size of the words. We have also noticed emotional variations among the people for the same news or events. For example, while many people considered lockdown as positive, there were another group of people who were against it. Therefore, the same words with a little variation changed the emotion of the tweets. Machine learning models are highly dependable on the quality of the data. Most of the models rely on good data annotation and embedding techniques. This encouraged us to create our own for emotion analysis on the ongoing pandemic. Further, to make a robust model that can adapt to the change of the emoticon and punctuation uses in the tweets, we have developed a deep learning model pipeline. In the following subsection, we briefly describe the process of data collection and data annotation along with an overview of our dataset.

3.1. DATA COLLECTION

We are collecting tweets since 5th March 2020 using Twitter Streaming API and the python Tweepy package. We have collected more than 500M tweets in 2020. We run the queries using COVID-19 related keywords (e.g. COVID, corona, coronavirus) for the tweets collection. The module listens to the stream of the tweets and tries to check if a tweet text contains any of the desired words. While checking the module it converts all the text to lower case and tries to find out sub-strings within the text. By doing this, the module identifies a qualified tweet and saves it in the JSON format. Further, the collected data is processed in real-time for the CoronaVis [5] application. We will keep collecting the data and update the collected tweets ids in the data repository [6] periodically. The repository contained those tweet ids for which we were able to estimate a state-level geo-location.

3.2. DATA ANNOTATION

We randomly selected 10K English tweets generated from the USA for the emotion annotation from the collected COVID-19 tweets in our first phase of data annotation. The tweets are annotated manually by 3 different people to reduce any bias. Among three annotators, one is a PhD student working on social media data mining since 2017. The other two annotators are undergraduate students from the computer science department and are native English speakers. We have selected 10 dominant emotions based on the study in [34] to label the tweets. Those 10 labels are neutral, optimistic, happy, sad, surprise, fear, anger, denial, joking, and pessimistic. Each tweet has annotated with primary and secondary emotion based on the tweet text. The primary label is selected from the majority agreement of all the annotators considering both primary and secondary labels. For example, if an annotator selected “Optimistic” as the primary label and another annotator selected “Optimistic” as a secondary label for a tweet, we have considered the primary label for that tweet as “Optimistic”. The secondary emotion is selected based on the majority agreement. If a majority agreement is unavailable, then that the tweet was discarded. By this process, the agreement for primary emotion between two annotators is 87% and the agreement from three annotators is 68%. For the secondary emotion, the inter-annotators agreements are 54% and 41% respectively by two annotators and three annotators. Further, the annotators marked a phrase associated with the primary emotion for each tweet. The whole tweet text has been selected for the tweets with neutral emotions. We will share our annotated emotion data publicly for further research and analysis.

[5] https://mykabir.github.io/coronavis/

[6] https://github.com/mykabir/COVID19

3.3. DATA DESCRIPTION

3.3.1. COVID-19 Tweets Data. Table 1 represents a high-level summary of the tweets ids that is available in the git repository. However, we are continuously collecting the data and thus the data statistics can be changed in the repository with future updates. In the repository, we have included processed tweet ids that have geolocation information. However, we will also include the list of all tweets ids with or without geo-information.

Table 1. COVID-19 Tweets Data Summary

Attribute	**Summary**
Collection Period	March 5, 2020 to December 31, 2020.
Number of unique tweets	56,014,158.
Location	USA (State label).
Number of Unique users	Total: 5,427,831; Verified: 56,387;

The processed tweets ids are saved and updated in the git repository within the folder named as data. The data folder contains several csv files. Every file contains tweets ids fetched in the respective date that is specified as the name of that file. For example, 2020-03-05.csv contains the tweets that was fetch on 5th March, 2020. The name was formatted as Year-Month-Date.

3.3.2. Annotated EMOCOV Data. Table 2 provides the label distributions of different types of emotions in the annotated datasets. We can see that there is a good variation in the label distribution. We can also see that a large number of tweets were annotated in the Surprised, Anger, and Neutral categories where there are only a few tweets in Denial and Joking categories.

Table 2. The label distributions in COVID-19 Annotated Emotion Dataset (%)

Type	**neu.**	**opt.**	**hap.**	**sad**	**sur.**	**fea.**	**ang.**	**den.**	**jok.**	**pes.**
Primary	23.47	8.43	8.29	7.82	16.64	8.79	16.83	1.16	4.64	3.93
Secondary	38.54	7.90	3.99	12.61	7.17	9.28	4.99	1.43	2.21	11.86

Table 3 presents a few examples of annotated tweets. The first emotion is the primary emotion and selected text represents the reasoning behind that emotion. Combining the labels from different annotators we decide the primary and secondary emotions. In Section 5.2.2, Table 12 contains few more examples of phrase selection by annotators where we discuss the performance of our model.

Table 3. Example of annotated tweets

Example Tweet and Selected Text	**Emotion Category**
Tweet: Relief provided to the poor needy during lockdown and to facilitate medical reserves to combat COVID Selected Text: Relief provided to the poor	Happy Optimistic
Tweet: In the Covid era mathematical models are deciding matters of life and death. @mathbabedotorg explains how they wor. . . Selected Text: mathematical models are deciding matters of life and death	Surprise Fear
Tweet: We pay an obscene amount of taxes in NY. We aren't broke bc of COVID. We are broke because #GovernorDeath puts illegals. . . Selected Text: We are broke	Anger Pessimistic

4. EMOTION DETECTION AND EXTRACTION

4.1. NEURAL NETWORK FOR EMOTION CLASSIFICATION

We develop a Deep Neural Network to classify the tweet text into a specific emotion category. To create the network, we modify the deep neural network that we have proposed in our previous research work [4]. Figure 2 illustrates the architecture of the model starting from input sequence generation. The modified deep learning model comprises 6 primary components.

1. Input layer: Processed tweets are used as input in this layer as vectors.

2. Embedding layer: Using lookup tables, this layer encodes the input into real-valued vectors. We used a pretrained word vectors named GloVe [42] which generates a feature word vectors using co-occurrences based statistical model. This layer map all tokenized words in every tweet to their respective word vector. Padding is used at the end of the vector list for the tweets with shorter length.

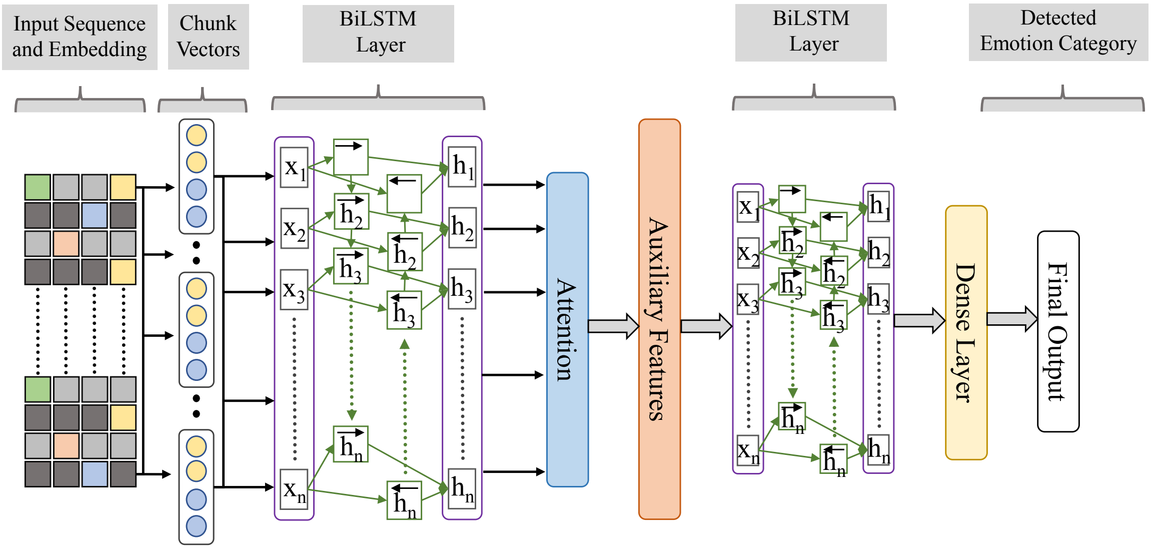


Figure 2. The illustration of the emotional classification Deep Neural Model

3. BLSTM layer: The Long-Short Term Memory (LSTM) is a specialized version of Recurrent Neural Network (RNN) that is capable of learning long term dependencies. While LSTM can only see and learn from past input data, Bidirectional LSTM runs input in both forward and backward direction. This bidirectional feature of BiLSTM is critical to understand of complex language context[43].

   The implemented LSTM version in this work can be defined by the equations 1-5 where the input gate i_t, forget gate f_t, output gate o_t, and cell state activation c_t. In the equations σ represents the logistic sigmoid function, h represents the respective hidden vectors, and W is the weight matrix. A detailed explanation of each equation and more about LSTM is available in [44].

$$i_t = \sigma \left(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i \right) \tag{1}$$

$$f_t = \sigma \left(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f \right) \tag{2}$$

$$o_t = \sigma \left(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o \right) \tag{3}$$

$$c_t = f_t c_{t-1} + i_t \tanh \left(W_{xc} x_t + W_{hc} h_{t-1} + b_c \right) \tag{4}$$

$$h_t = o_t \tanh \left(c_t \right) \tag{5}$$

4. Attention layer: We use a word-level deterministic, differentiable attention mechanism to identify the words with the closer semantic relationship in a tweet. Equation 6 represents the attention score $e_{i,j}$ of each word t in a sentence i and g is an activation function. More information on the attention mechanism is available in [45].

$$e_{i,j} = g \left(W h_t c \right) \tag{6}$$

5. Auxiliary features input: A tweet can only contain 280 characters which forces a user to express emotions in a different way compared to a traditional English sentence. People use extra punctuations and emoticons to intensify the meaning of a tweet. We perform feature engineering to obtain a set of specific auxiliary features that can assist the classification model. A list of extracted auxiliary features that shows noticeable influence during the model evaluation is given in Table 4. The well-known NLTK package is used to extract those features.

Table 4. Auxiliary Features

polarity, subjectivity, wordsVsLength, digitVsLength, punctuationVsLength, nounsVsWwords, sadVsWords, capitalsVsWords, uniqueWords, TagNumbers.

6. Output layer: The output layer is created using dense layers which use *sigmoid* as activation function and predict the output class. The layer produces binary values for all the label categories.

Classification Model Parameters and Training A set of optimal parameters is crucial for achieving the desired performance results. We performed rigorous parameter tuning and selected an optimal set that is used in all the experiments. We used the same set of parameters as presented in Table 5 for performance evaluation and model reproducibility. To build a robust model, we used 5-fold cross-validation with an 80/20 split ratio for training and testing. Initially, we have trained and tested our model starting from 20 epochs to 100 epochs. To get the optimal learning rate, we employed an LR-scheduler with an initial learning rate of 0.001. We observe that the learning rate drops to 0.00001 by the time the model reaches the best validation score. We noticed that each model performed best around 40 epochs and after that start overfitting. Therefore we use 50 epochs for the final training and testing.

Table 5. Hyperparameter values

Hyperparameter	**Value/Description**
Text embedding	Dimension: 250
BLSTM Layer	2 layers; 250 hidden units in each (Forward and Backward)
Dense Layer	3 layers; First 2 layers have 150 and 75 units respectively and the last one is output (Dense)
Drop-out rate	Word Embedding: 0.3; Dense layer: 0.2 each;
Activation function	Conv1D, BLSTM, Dense: ReLU; Output Dense layer: Sigmoid;
Adam optimizer	Learning rate 0.001-0.00001; $beta_1$=0.8;
Validation	Training and Validation Split = 80/20;
Epochs and batch	Epochs = 50; batch size = 68;

4.2. CUSTOM ROBERTA FOR PHRASE EXTRACTION

RoBERTa, a Robustly Optimized BERT Pretraining Approach [40] is developed using the Google's BERT language masking strategy [46]. The accuracy of RoBERTa is 2-20% higher compared to BERT. Both of these approaches provide transformer to learn a language representation. However, BERT is more suitable for Question and Answer problem solution as BERT tries to predict the Next Sequence Probability of a token. As RoBERTa does not use NSP, we have to develop a custom Q&A head to predict the probability of the start and the end sequence of the tokens.

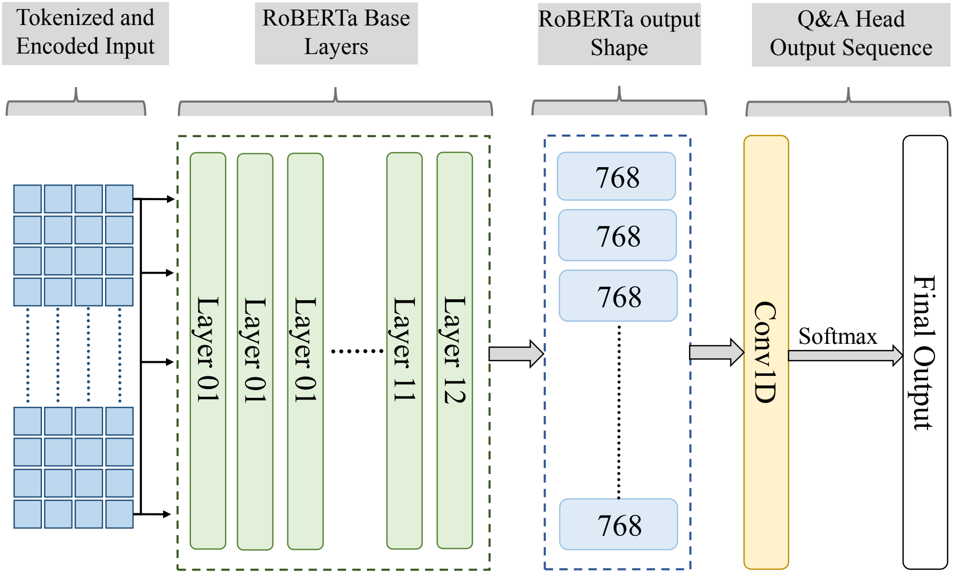


Figure 3. RoBERTa model illustration with custom Q&A head

The developed model uses two Q&A heads that is illustrated in Figure 3 for the position of start sequence of a phrase and the end position of the sequence. Practically, the model provides a probability for each character position for being a start or end sequence. Further, using the maximum probability value, the model selects the final start and the end positions. Primarily, the developed models have the following three components:

1. Tokenizer: The tokenizer takes the input text and split it into words using the black space between the characters. It performs the similar split for both the input tweet text and the selected text. Further, it translates those words into the respective numerical values using RoBERTa base vocabulary files. After that it creates two masked lists where any other values apart from the start and end positions of the sequence is set as 0. The formal list puts 1 for the start position and the second list puts 1 for the end position of the selected text sequence. The tokenizer also creates a same size attention mask for all the input tweet texts where available words position presented as ones with the padding zeros.

2. RoBERTa Base: We used pretrained RoBERTa base model for further training with our input data. RoBERTa base uses the BERT-base architecture with 125M parameters. For the implementation, we used Huggingface transformer library. Detailed information about RoBERTa base is available in Liu et al. [40].

3. Custom Q&A head: We created a custom Q&A head for the start and the end position prediction of the sequence. RoBERTa is developed primarily for question and answering task. In our model, we treated it for the similar purpose where the emotion label is the question, and the selected phrase is the respective answer. To achieve that, we use a convolution layer that transform the base output of RoBERTa to a pre-determined vector size. Further, applying the softmax function, it produces two one hot encoded lists for the starting and the ending index position of the given text.

5. EXPERIMENTAL RESULT AND ANALYSIS

We present the experimental result and historical emotion analysis in this section. We use two different machines to perform data collection, model training, and analysis. We use a machine with Intel Xeon E5-2650 v4 @ 2.20GHz CPU (12 cores, 24 threads) with 64GB RAM and an Nvidia RTX-2070 super GPU. Another machine comprises of Intel® Core™ i9-9900K CPU @ 3.60GHz (8 cores, 16 threads) with 64GB RAM and an Nvidia RTX-2080Ti GPU. In the following subsections, we describe the evaluation metrics, experimental results, and emotion analysis.

Model Parameters We used several parameters to tune our model. Table 6 demonstrates the final parameter for our model with the best performance.

Table 6. Hyperparameter values

RoBERTa Hyperparameter	**Value/Description**
MAX Input Length	196
Pre-trained Network	RoBERTa Base
Dense layer	2 layers; One for start position and one for end position of the sequence.
Dropout	0.1 before each output Dense layer
Activation function	Output Dense layer: Softmax;

Cross Validation	Folds = 5;
	Training and Validation Split = 80/20;

Epochs and batch	Epochs = 10 (each fold); batch size = 68;

5.1. EVALUATION METRICS

To evaluate the classification model, we have used Micro F1, Macro F1, Jaccard, and Accuracy. Let L denotes the number of label categories, TP denotes True Positive, FP denotes False Positive, and FN denotes False Negative. We can define F1 micro average score using equations 7-9.

$$Precision_{micro} = \frac{\sum_{k=1}^{L} TP_k}{\sum_{k=1}^{L}(TP_k + FP_k)} \tag{7}$$

$$Recall_{micro} = \frac{\sum_{k=1}^{L} TP_k}{\sum_{k=1}^{L}(TP_k + FN_k)} \tag{8}$$

$$F1_{micro} = \frac{2 * Precision_{micro} * Recall_{micro}}{Precision_{micro} + Recall_{micro}} \tag{9}$$

Equations 10-13 denote the macro average F1 score calculation which is a simple averaging of F1 scores for different labels.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

$$F1_{macro} = \frac{1}{|L|} \sum_{k=1}^{L} F1_k \tag{13}$$

Jaccard score is a popular metrics for multi-label binary classifier accuracy as this metric consider every label category similarly. The jaccard score is calculated using the equation 14. In the equation, T denotes the number of test data, Y_k denotes the truth label of data k, and P_k denotes the predicted label.

$$Jaccard = \frac{1}{|T|} \sum_{k=1}^{T} \frac{Y_k \cap P_k}{Y_k \cup P_k} \tag{14}$$

$$Accuracy = \frac{1}{T} \sum_{k=1}^{T} \sigma(Y_k == P_k) \tag{15}$$

We used accuracy as another metrics as it provides a better observation of model performance while the data has imbalanced categories. Equation 15 defines the Accuracy score where $\sigma(Y_k == P_k)$ returns 1 if the prediction for a data is correct, otherwise it returns 0.

We evaluated phrase extraction model using word-level jaccard similary score. It calculates the performance using the similarity between the predicted words respective ground truth. In equation 16, Y_k denotes the ground truth string, and P_k refers to the predicted string.

$$Jaccard_{similarity} = \frac{1}{|T|} \sum_{k=1}^{T} \frac{len(Y_k \cap P_k)}{len(Y_k) + len(P_k) - len(Y_k \cap P_k)} \tag{16}$$

5.2. EXPERIMENTAL RESULTS

5.2.1. Classifier Evaluation. We evaluate our proposed classification model using two different data sets. First, we use our own labeled emotion data that we described in Section 3.3.2. The primary and secondary emotion label was processed as distinctive data points for the classification purpose. For example, if a tweet has a label of Angry and Pessimistic, we use that tweet for both of the label categories individually. Further, we have created an aggregated data combining our data, the emotion dataset by Yang et al. [11] and the emotion classification dataset of SemEval-2018 Task1: Affect in Tweets [10]. For the aggregated data, we evaluate the models only on the selected labels that are similar across all the three datasets. We have converted some of the labels to reduce the imbalance in the label categories.

Table 7. The label distributions in combined Emotion Dataset (%)

		neu.	**opt.**	**hap.**	**sad**	**sur.**	**fea.**	**ang.**	**jok.**	**pes.**
# data		10K	25K	15K	25K	25K	15K	15K	20K	25K
% label	**neu.**	23.47	7.18	11.12	7.18	7.18	11.12	11.12	10.77	7.18
% label	**opt.**	8.43	15.35	13.03	15.35	15.35	13.03	13.03	14.33	15.35
% label	**hap.**	8.29	9.83	15.23	9.83	9.83	15.23	15.23	3.89	9.83
% label	**sad**	7.82	14.66	13.05	14.66	14.66	13.05	13.05	13.19	14.66
% label	**sur.**	16.64	11.75	9.96	11.75	11.75	9.96	9.96	16.04	11.75
% label	**fear**	8.79	6.40	9.91	6.40	6.40	9.91	9.91	4.16	6.40
% label	**ang.**	16.83	12.72	19.71	12.72	12.72	19.71	19.71	7.93	12.72
% label	**jok.**	4.64	14.67	2.47	14.67	14.67	2.47	2.47	22.02	14.67
% label	**pes.**	3.93	7.43	5.52	7.43	7.43	5.52	5.52	7.67	7.43

The aggregation module produces a combined dataset across 9 different emotion categories that are presented in Table 7. The #$data$ row in the table presents the number of tweets that are used for training, validation, and testing of the classification models for each category. Other rows indicated as %$label$ represents the distribution of labeled data in each dataset. To elaborate, to train and test the models to identify the neutral tweets we used a dataset containing 10K tweets where 23.47% tweets were neutral and the rest of the tweets were labeled as other emotion categories. To train and test, those 23.47% tweets were assigned binary label 1 - indicating neutral emotion, while the rest of the 77.53% tweets was assigned label 0 - indicating non-neutral tweets. Similarly, for optimistic we used a dataset containing 25K tweets, where 15.35% tweets were optimistic and rest of the tweets were labeled as other categories.

Table 8. Classifier Evaluation and Comparison

Model	**F1-Micro**	**F1-Macro**	**Jaccard**	**Accuracy**
SVM-Unigrams	0.5294	0.4076	0.4138	0.7383
NTUA-SLP	**0.5981**	0.4887	**0.5472**	0.8492
BiLSTM$_{AAf}$(Our)	0.5514	**0.5392**	0.5366	**0.8647**

Tables 8 and 9 represent the performance of 3 different classification models. To compare our model performance, we compare our model with SVM-Unigrams [10] and NTUA-SLP [37]. NTUA-SLP is the submitted system that became the winner of the SemEval-2018 Task1: E-cchallenge. For our annotated dataset which has highly imbalanced categories, NTUA-SLP performed better in F1-Micro and Jaccard score. However, our model performed better in the other two metrics. Our model BiLSTM$_{AAf}$ outperforms both SVM-Unigrams and NTUA-SLP in terms of F1-Macro, Jaccard, and Accuracy while we train and test those models using the combined dataset described in Table 7.

Table 9. Classifier Evaluation and Comparison using combined emotion data

Model	**F1-Micro**	**F1-Macro**	**Jaccard**	**Accuracy**
SVM-Unigrams	0.5532	0.4849	0.5185	0.8227
NTUA-SLP	**0.7058**	0.5829	0.6293	0.8746
BiLSTM$_{AAf}$(Our)	0.6893	**0.6342**	**0.6475**	**0.8951**

Table 10 represents some sample tweet texts and respective emotion labels predicted by our proposed model (BiLSTM) and NTUA-SLP. We omit SVM-Unigrams from this comparison as the performance of this model is considerably lower. Although both models predicted labels for all of the emotion classes for a given text, here we only present the emotion labels for which the models have different predictions. Column 'GT' in the table denotes the ground truth (annotated) labels.

Table 10. Sample output comparison between the proposed model and NTUA-SLP model

	Tweet Text	**Emotion**	**BiLSTM$_{AAf}$**	**NTUA-SLP**	**GT**
1	Those who are following trump regarding MASK, have a happy get together. #covid #ignorant	Happy	0	1	0
		Anger	1	0	1
2	With all of the sad news during COVID, the only hopeful thing is Stimulus check for the struggling family.	Sad	1	1	1
		Optimistic	1	0	1
3	If this #lockdown does not end now it won't be just the covid that is flattened but the economy FLATLINED.	Sad	1	1	1
		Anger	1	0	1
4	Plans to Alter Own Clothes After Losing 17 Pounds in COVID-19 #Lockdown	Happy	1	0	1
		Optimistic	1	0	1
5	This is nothing more than targeting the old to get increased numbers of deaths with COVID. OBVIOUS!	Pessimistic	0	1	1
		Anger	1	0	1

We observe that NTUA-SLP is struggling with sarcastic and contrasting emotions. For example, in the first tweet, the tone of the text seems happy until we see the word #ignorant. Due to this hashtag, we can infer that this tweet is sarcastic. In the second and third tweets, we observe contrasting emotions or meanings. While the struggle of the families during covid is sad, stimulus check brings optimism. In the third tweet, the literal meaning of the word 'Losing' is not something positive. However, losing weight could be a positive thing. We find it fascinating that our proposed model is doing well do identify these contexts compared to the NTUA-SLP. To find out the probable reason behind this we perform several evaluations. In the evaluation, we observe the impact of auxiliary feature input that we describe in Section 4.1. Using auxiliary features input we explicitly provide a set of features that helps the model to detect the contrast in the tweet. For example, in Table 10, we observe a significant number of capital words or letters in the tweets with contrasting meanings. The auxiliary features input helps the model to catch this information which is otherwise might have less impact due to the attention on the words and word-embedding. By the architecture, NTUA-SLP uses a self-attention mechanism that identifies the dominant words related to the emotion. However, this leads to misclassification in some cases. To confirm this hypothesis we further train and evaluate our model without using the auxiliary features input. Without the auxiliary features, the performance of the model drops by 5-10% for different emotion classes. Furthermore, we assess the weakness of our model. Our proposed model underperforms for the emotion classes with small training data such as pessimistic and fear. The 5th tweet in the table represents such an example. NTUA-SLP is outperforming our model for such a situation. In the future, we are planning to develop and train multiple models architecture with and without auxiliary features and ensemble those models to address the weakness of our model.

5.2.2. Phrase Extraction Evaluation. Similar to classifier evaluation we evaluate models for phrase extraction using our annotated dataset and a combined dataset that is available in Kaggle Tweet Sentiment Extraction competition [7]. However, due to some automated data processing, there were some issues in the text in the available dataset. We processed that dataset using the original tweet text that is available in crowdflower dataset [8]. Combing our data with the external dataset, we were able to make the models robust and it increased the performance of the models. Table 11 represents the performance evaluation

[7] https://www.kaggle.com/c/tweet-sentiment-extraction/

[8] https://data.world/crowdflower/sentiment-analysis-in-text

of the models. In the table Jaccard (EXT.) denotes the performance of the model when we also used the external data for model training and testing. The developed RoBERTa model with a custom Q&A head outperforms both BERT and ALBERT models for both datasets. For BERT and ALBERT implementation, we have used the publicly available top kernels used and available in the Kaggle Tweet Sentiment Extraction competition.

Table 11. Performance Evaluation of the Phrase Extractors

Model	**Jaccard**	**Jaccard (EXT.)**
BERT Base	0.6852	0.7349
ALBERT	0.6879	0.7529
Custom RoBERTA (Our Model)	**0.7196**	**0.7865**

Few examples of phrase extractions are presented in Table 12 to demonstrate the effectiveness of the model in different contexts. The table also includes the output from BERT and ALBERT models along with our proposed Custom RoBERTa model. We observe that in most cases, all of the models selected smaller phrases or fewer words compared to the annotators' selection. However, our proposed model selected longer phrases in many cases compared to other models. All three models follow the similar concept of question and answer modeling. In the context of this work, the provided emotion acts like a question and the answer is the selected phrase by the models related to the given emotion. Both BERT and ALBERT encode each word in a tweet and selected text. However, we created a custom head in our model which encodes each letter in the text instead of the word. Hence, while BERT and ALBERT try to predict the starting and ending word positions, our model tries to predict the starting and ending letter positions. We believe this behavior is the primary reason for the better performance of our model as it helps to mimic the longer phrase. In Table 12, we observe both BERT and ALBERT are omitting the preposition, adverbs, or adjectives in the predicted text in many cases. For example, both models omitted "should, biggest, have, and some" for tweets 1-4. while our proposed model included those words.

In the 5th tweet, our model predicted 'What kind of' compared to the 'What kind' predicted by ALBERT.

Table 12. Example of Phrase Extractions by Proposed Model

	Example of Phrase Extraction
1	Tweet: Almost 70% of PA's Covid-19 deaths 2611 of 3806 have occurred in nursing homes or long-term care — PA should never have. . . Emotion: Anger, Selected Text: pa should never have **BERT Base:** never have, **ALBERT:** never have **Custom RoBERTa:** should never have
2	Tweet: Rare Thai Turtle Nests Make Biggest Comeback In 20 Years Thanks to COVID-19 Emotion: Happy, Selected Text: biggest comeback in 20 years **BERT Base:** comeback, **ALBERT:** thanks **Custom RoBERTa:** biggest comeback
3	Tweet: If hygiene JUST became a priority for you ... you have bigger issues than Corona. Emotion: Pessimistic, Selected Text: bigger issues than Corona **BERT Base:** bigger issues, **ALBERT:** bigger issues **Custom RoBERTa:** have bigger issues
4	Tweet: Fight Corona by staying indoors. Spend some quality time with your family that is otherwise difficult in our busy schedules. Emotion: Optimistic, Selected Text: Spend some quality time **BERT Base:** quality, **ALBERT:** quality time **Custom RoBERTa:** some quality time
5	Tweet: He believes the Democrats want people to die of COVID-19 so they can win the election? What kind of hatred is in his heart!! Emotion: Surprise, Selected Text: What kind of hatred is in his heart!! **BERT Base:** election? What, **ALBERT:** What kind **Custom RoBERTa:** What kind of

The research on the phrase extraction models is still in the primary stage for emotion context. Also due to the subjectivity of the annotators, the selected text varies a lot. The models perform miserably with fear, surprise, and sarcastic tweets. In future, we need to conduct more experiments and analysis to have more concrete reasoning on why the models performing differently. Also, we need more data for generalizing the models better.

5.3. HISTORICAL TWEETS EMOTION ANALYSIS

In this section, we present the historical emotion analysis on the COVID-19 tweets. We present the analysis of the six dominant emotions (e.g. Happy, Sad, Optimistic, Pessimistic, Fear, and Anger) all over the USA. Further, we analyze the emotions of six individual states (NY, CA, CO, TX, MO, and FL) to perform a comparative study of the emotions among the states from the east coast, midwest, and west coast. To infer the state from the tweet we have used geo-tag and user profile information. If a tweet is not geotagged, we fetched the user profile to lookup the location info. We discarded the tweets if we were unable to infer a location. We have also discarded tweets from any user profile which has more than 5 tweets on a day. This is to ensure the filtering of the spamming and also reducing the bias of having tweets from the same person. We have also removed the duplicates or retweets. Using our location detection strategy and filtering module, we get more than 56M tweets originated from the USA from 5th March 2020 to 31st December 2020. On average there are 188765 tweets per day. For the above specified six states that we have used for the analysis have the following numbers tweet per day on average: NY-11419, CA-24230, CO-3681, TX-19328, MO-2297, FL-13014. For the analysis, we use our proposed machine learning model to classify the tweet emotions. In this section, we include analysis on weekly and monthly emotion distribution. However, we primarily focus on the monthly analysis at which enables us to correlate the important events during the pandemic in limited space. To calculate the emotion scores in the figures, we use the weekly and monthly mean of the classified tweets emotions.

Figure 4 presents the weekly ratio of emotion categories. We can see that happy, sad, and fear are the identified emotions for most of the tweets. We observe that while 70-80% of the tweets are showing those 6 emotions, there are still 20-30% tweets that are either neutral or can be categorized in other emotion categories. In the figure, Y-axis represents the distributions of emotions on a scale of 0 to 1. The distribution is calculated using the total number of tweets identified for an emotion divided by the total number of tweets in that periods. For example, in the first week the distribution of the emotions are as follows: Happy = 0.1714, sad = 0.1477, optimistic = 0.1394, pessimistic = 0.0589, fear = 0.1537, anger = 0.0298, surprise = 0.0092, and others = 0.2899.

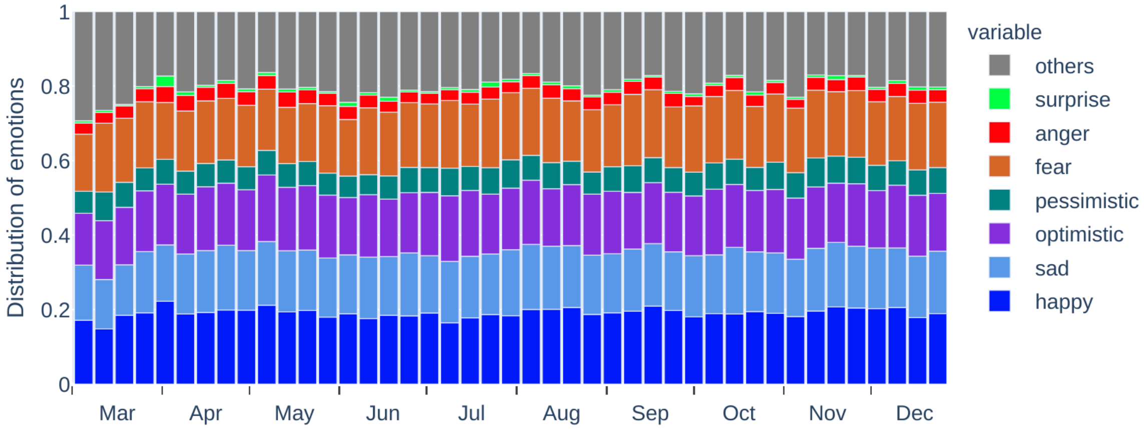


Figure 4. Weekly emotion distribution in the USA

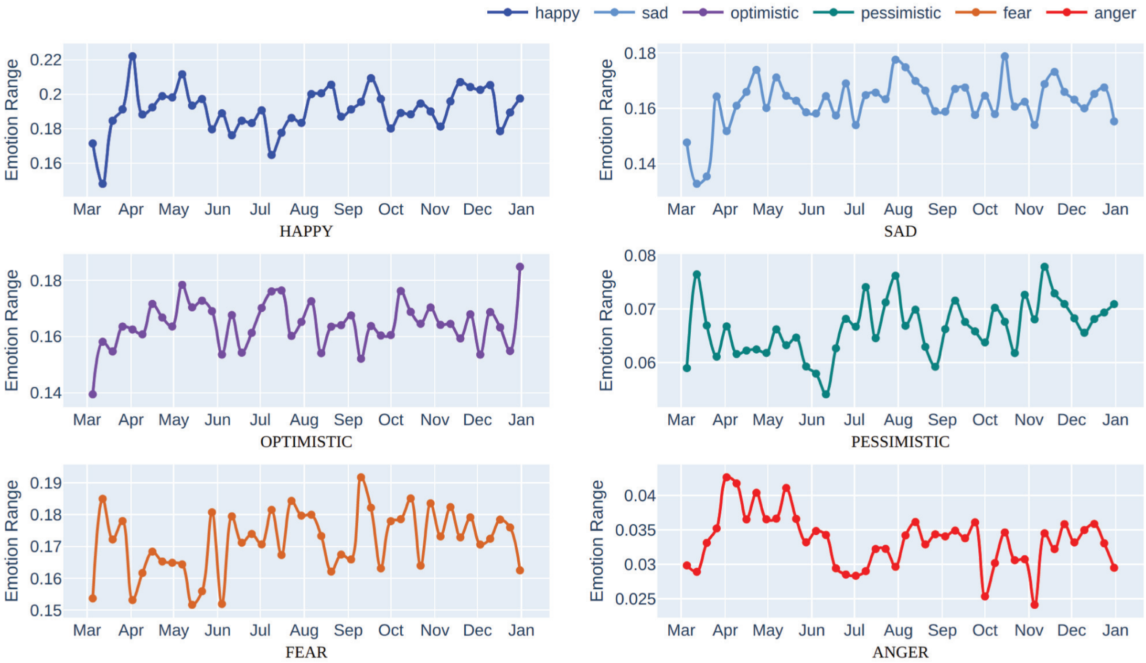


Figure 5. Weekly emotion variation in the USA (March 2020 - December 2020). Y axis represents the weekly emotion range on a scale of 0-1 combining all emotions.

We conduct further emotion analysis on the six dominant emotions that we have stated earlier. Figure 5 provides a better idea of weekly emotion distribution. It shows the variation in the emotions in each week. We use the exact emotion range in the Y-axis without scaling. This allows us to recognize the dominant emotions in the tweets. For example, the Y-axis values of pessimistic and anger charts denote that the number of tweets with those emotions is lower than other emotions. While Figure 5 represents the emotional roller-coaster in the USA, Figure 6 depicts a better picture of emotion evolution during the pandemic using monthly emotion distribution. We can observe a similar emotion range in monthly and weekly charts. We present some of the critical events during the pandemic in Figure 7 to correlate the emotions. This also allows us to observe the accuracy of the models with respect to historical events. In Figure 7, the events are ordered in a way such that, closer events to the timeline occurred earlier in the respective month. From the figure, we can see that in mid-February US stock market crashed from the fear of COVID-19. By the end of February US reported the first COVID related death.

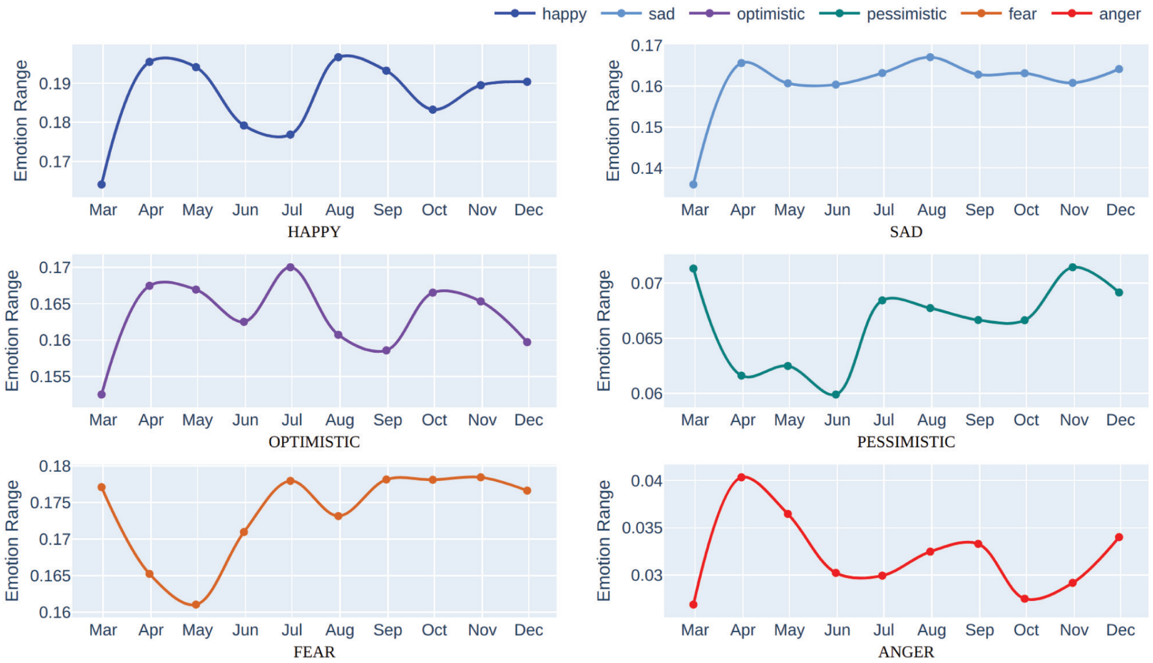


Figure 6. Monthly emotion variation in the USA (March 2020 - December 2020).

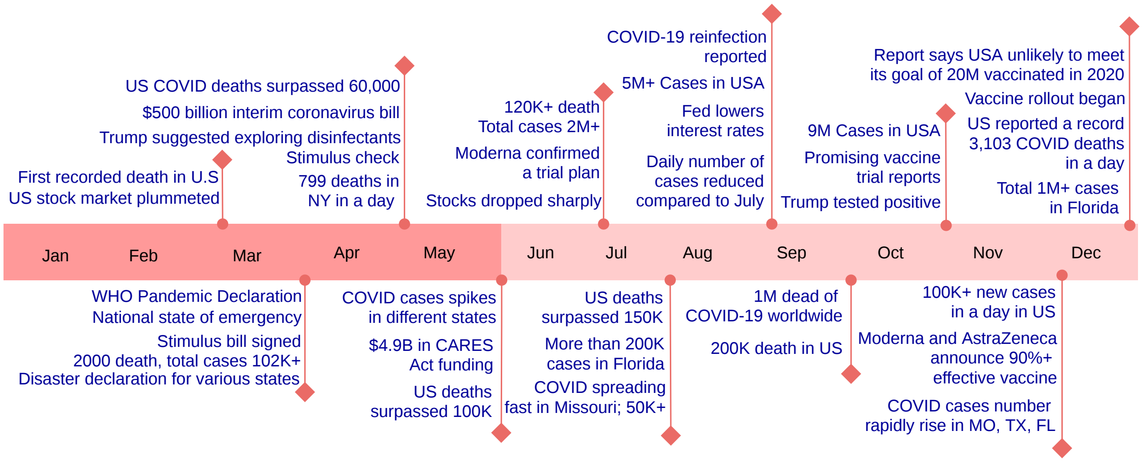

Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
First recorded death in U.S
US stock market plummeted
US COVID deaths surpassed 60,000
$500 billion interim coronavirus bill
Trump suggested exploring disinfectants
Stimulus check
799 deaths in NY in a day
120K+ death
Total cases 2M+
Moderna confirmed a trial plan
Stocks dropped sharply
COVID-19 reinfection reported
5M+ Cases in USA
Fed lowers interest rates
Daily number of cases reduced compared to July
9M Cases in USA
Promising vaccine trial reports
Trump tested positive
Report says USA unlikely to meet its goal of 20M vaccinated in 2020
Vaccine rollout began
US reported a record 3,103 COVID deaths in a day
Total 1M+ cases in Florida
WHO Pandemic Declaration
National state of emergency
Stimulus bill signed
2000 death, total cases 102K+
Disaster declaration for various states
COVID cases spikes in different states
$4.9B in CARES Act funding
US deaths surpassed 100K
US deaths surpassed 150K
More than 200K cases in Florida
COVID spreading fast in Missouri; 50K+
1M dead of COVID-19 worldwide
200K death in US
100K+ new cases in a day in US
Moderna and AstraZeneca announce 90%+ effective vaccine
COVID cases number rapidly rise in MO, TX, FL

Figure 7. Timeline of Events Related to the COVID-19 Pandemic in the USA

From the emotion chart in Figure 6 we observe the high range of fear and pessimism at the beginning of March as people became aware of the situation. In March WHO declares COVID-19 as a pandemic and a national emergency also announced in the USA. By the end of March, the death count became 2000 in the US and the total number of cases surpassed 102K+. However, stimulus bills were also signed in March and people started to receive their first stimulus check in April. There was also a lack of proper guidance regarding the pandemic and many people thought COVID-19 is only harming the adult people severely. Because of this, the fear is reduced and people became optimistic in April. However, people were still sad and disappointed by the pandemic and economic situation. We can see a sharp rise in anger in April. In April, the death count increased rapidly and president Trump suggested disinfectants can be helpful for COVID treatment which surges the anger among the people. Until May, most of the COVID cases in the USA were came from New York. However, by the end of May, COVID cases and hospitalization started to spike in other states which triggers negative emotions all over the USA. This reflects in Figure 6 as we can see fear and pessimism rise sharply from June. In August, the daily reported new cases declined and because of that, we see a drop in the fear. People were scared again after August as the second wave of COVID infection started and the daily new cases started to break the previous record regularly. By September 200K people died in the USA and a total of 1M people died worldwide because of COVID. In October several reports were published about positive vaccine trials which gave optimism to the people. In November, US reported 100K+ news cases in a single day. People started to lost hope and both anger and pessimism started to rise. In December people started to gain confidence because of the vaccine roll-out. However, the USA experienced a record single-day death. Furthermore, several reports stated the 20M vaccination goal of the USA might not be fulfilled in 2020. All those events trigger mixed reactions but we can observe an increasing amount of anger.

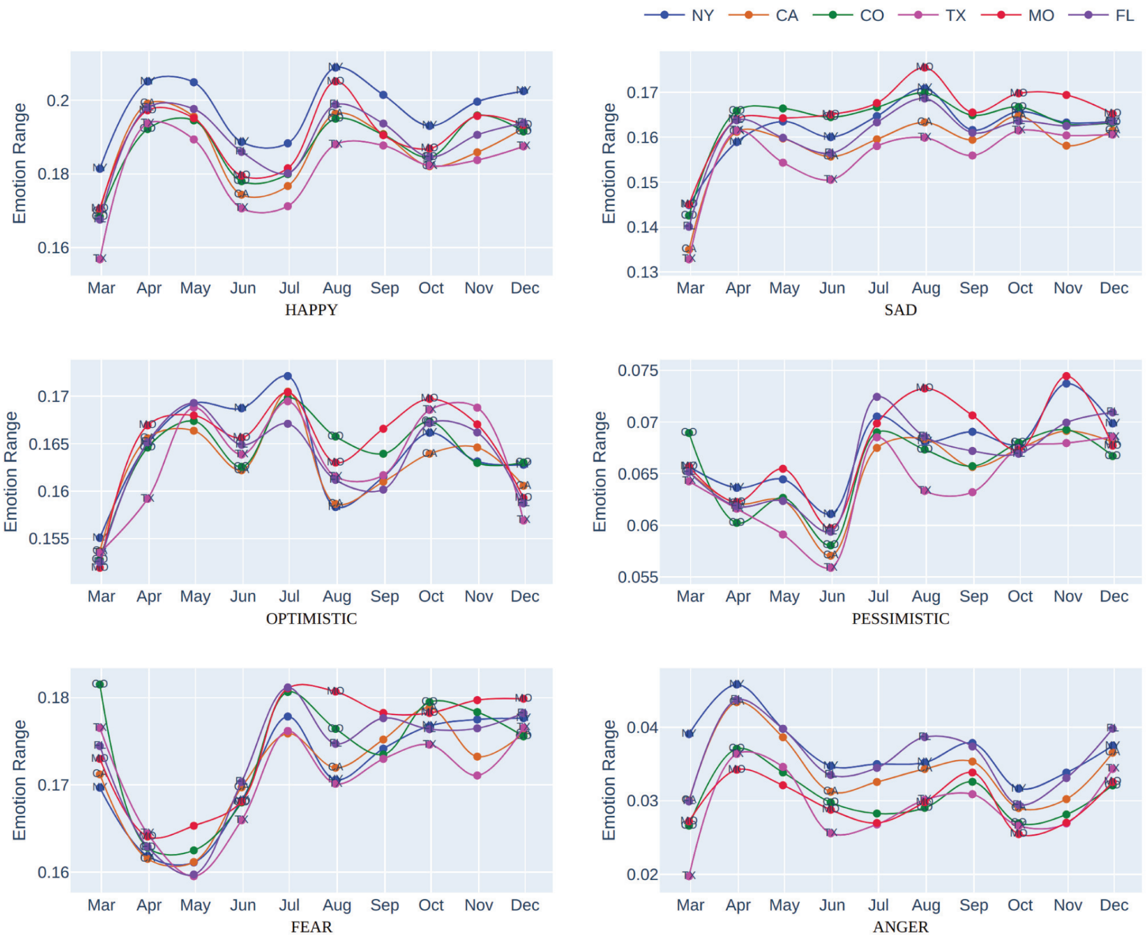


Figure 8. Monthly emotion variations in 6 states of USA (March-December 2020)

The monthly emotions variations for six different states (NY, CA, CO, TX, MO, FL) are depicted in Figure 8. We can observe a similarity in emotion timeline across the USA and the states. While most of the states have similar emotional trends, we can observe some significant variations at some points. For instance, we can observe a higher amount of negative emotions such as fear, pessimism, and sadness in Missouri (MO) and Florida (FL) during July, August, and September. MO exhibits a higher amount of fear, pessimism, and sadness in August compared to other states. If we look back at the timeline of the events in Figure 7, we see that in July COVID-19 cases spiked in MO and FL and it was spreading fast. This correlates with the higher negative emotion as we can see in the chart. In November and December, the new cases again started to rise rapidly in MO and FL which make people scared and sad. As a result, we can see those states showing high fear and pessimism. We can see during NOV-DEC, MO is showing the highest fear among the six states and FL is showing maximum anger. From the charts and COVID-19 events timeline, we can state that the classification model performed satisfactorily to identify the emotions during the pandemic.

6. CONCLUSION AND FUTURE WORK

In this work, we proposed two machine learning models for multi-label binary classification and phrase extraction applied on a unique emotion dataset on COVID-19 tweets for classifying 10 different emotion labels, and to select a phrase that represents each emotion the most. This paper also presents a comparative performance evaluation and analysis of the proposed models. Our developed models outperformed other systems under different performance metrics. We use a set of auxiliary features that improve the performance of the classifier. For phrase extraction, we use RoBERTa pre-trained model with a custom Q&A head which takes the emotion label as a question and tries to find a phrase that can best be suited for that emotion. The output analysis of the model shows the robustness to understand the context of a given tweet. Further, we perform a historical emotion analysis over some of the states in the USA using the COVID-19 tweets. The analysis shows how the negative emotions increased during the pandemic. It also shows how people were adapting to the pandemic over time, and being more optimistic. In the future, we will integrate our models in our live application to continue the emotion analysis during the pandemic over the entire USA. We will also analyze phrase extraction model output over the historical COVID-19 tweets and incorporate those in the live application. We will keep exploring the different ideas on phrase extraction for emotion context in the future to improve our results further. We plan to use data augmentation and transfer learning to train our model so that it can perform robustly with effectiveness. We will share our data publicly for the different research communities on Github.

ACKNOWLEDGEMENTS

This research has been partially supported by a grant from NSF, CNS-1461914.

REFERENCES

[1] Xinyu Chen, Youngwoon Cho, and Suk Young Jang. Crime prediction using twitter sentiment and weather. In *2015 Systems and Information Engineering Design Symposium*, pages 63–68. IEEE, 2015.

[2] Matthew S Gerber. Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61:115–125, 2014.

[3] Purva Grover, Arpan Kumar Kar, Yogesh K Dwivedi, and Marijn Janssen. Polarization and acculturation in us election 2016 outcomes–can twitter analytics predict changes in voting preferences. *Technological Forecasting and Social Change*, 145:438–460, 2019.

[4] Md Yasin Kabir and Sanjay Madria. A deep learning approach for tweet classification and rescue scheduling for effective disaster management. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 269–278, 2019.

[5] Md Kabir, Sanjay Madria, et al. Coronavis: A real-time covid-19 tweets analyzer. *arXiv preprint arXiv:2004.13932*, 2020.

[6] Ansari Fatima Anees, Arsalaan Shaikh, Arbaz Shaikh, and Sufiyan Shaikh. Survey paper on sentiment analysis: Techniques and challenges. *EasyChair2516-2314*, 2020.

[7] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8 (4):e1253, 2018.

[8] Xingyou Wang, Weijie Jiang, and Zhiyong Luo. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437, 2016.

[9] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[10] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17, 2018.

[11] Qiang Yang, Hind Alamro, Somayah Albaradei, Adil Salhi, Xiaoting Lv, Changsheng Ma, Manal Alshehri, Inji Jaber, Faroug Tifratene, Wei Wang, et al. Senwave: Monitoring the global sentiments under the covid-19 pandemic. *arXiv preprint arXiv:2006.10842*, 2020.

[12] Jia Xue, Junxiang Chen, Chen Chen, ChengDa Zheng, and Tingshao Zhu. Machine learning on big data from twitter to understand public reactions to covid-19. *arXiv preprint arXiv:2005.08817*, 2020.

[13] Caleb Ziems, Bing He, Sandeep Soni, and Srijan Kumar. Racism is a virus: Anti-asian hate and counterhate in social media during the covid-19 crisis. *arXiv preprint arXiv:2005.12423*, 2020.

[14] Emily Chen, Kristina Lerman, and Emilio Ferrara. Covid-19: The first public coronavirus twitter dataset. *arXiv preprint arXiv:2003.07372*, 2020.

[15] Juan M Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, and Gerardo Chowell. A large-scale covid-19 twitter chatter dataset for open scientific research–an international collaboration. *arXiv preprint arXiv:2004.03688*, 2020.

[16] Pablo Martí, Leticia Serrano-Estrada, and Almudena Nolasco-Cirugeda. Social media data: Challenges, opportunities and limitations in urban studies. *Computers, Environment and Urban Systems*, 74:161–174, 2019.

[17] Patric R Spence, Kenneth A Lachlan, and Adam M Rainear. Social media and crisis research: Data collection and directions. *Computers in Human Behavior*, 54:667–672, 2016.

[18] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Coling 2010: Posters*, pages 36–44, 2010.

[19] Ruchit Nagar, Qingyu Yuan, Clark C Freifeld, Mauricio Santillana, Aaron Nojima, Rumi Chunara, and John S Brownstein. A case study of the new york city 2012-2013 influenza season with daily geocoded twitter data from temporal and spatiotemporal perspectives. *Journal of medical Internet research*, 16(10):e236, 2014.

[20] Mark Dredze, David A Broniatowski, and Karen M Hilyard. Zika vaccine misconceptions: A social media analysis. *Vaccine*, 34(30):3441, 2016.

[21] M Yasin Kabir, Sergey Gruzdev, and Sanjay Madria. Stimulate: A system for real-time information acquisition and learning for disaster management. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 186–193. IEEE, 2020.

[22] Lei Zou, Nina SN Lam, Shayan Shams, Heng Cai, Michelle A Meyer, Seungwon Yang, Kisung Lee, Seung-Jong Park, and Margaret A Reams. Social and geographical disparities in twitter use during hurricane harvey. *International Journal of Digital Earth*, 12(11):1300–1318, 2019.

[23] Zhou Yang, Long Hoang Nguyen, Joshua Stuve, Guofeng Cao, and Fang Jin. Harvey flooding rescue in social media. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2177–2185. IEEE, 2017.

[24] E Hirata, MA Giannotti, APC Larocca, and JA Quintanilha. Flooding and inundation collaborative mapping–use of the crowdmap/ushahidi platform in the city of sao paulo, brazil. *Journal of Flood Risk Management*, 11:S98–S109, 2018.

[25] Cody Buntain, Jennifer Golbeck, Brooke Liu, and Gary LaFree. Evaluating public response to the boston marathon bombing and other acts of terrorism through twitter. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[26] Brian G Southwell, Jeff Niederdeppe, Joseph N Cappella, Anna Gaysynsky, Dannielle E Kelley, April Oh, Emily B Peterson, and Wen-Ying Sylvia Chou. Misinformation as a misunderstood challenge to public health. *American journal of preventive medicine*, 57(2):282–285, 2019.

[27] Sunday Oluwafemi Oyeyemi, Elia Gabarron, and Rolf Wynn. Ebola, twitter, and misinformation: a dangerous combination? *Bmj*, 349:g6178, 2014.

[28] Zheye Wang, Nina SN Lam, Nick Obradovich, and Xinyue Ye. Are vulnerable communities digitally left behind in social responses to natural disasters? an evidence from hurricane sandy with twitter data. *Applied geography*, 108:1–8, 2019.

[29] Daniel Wladdimiro, Pablo Gonzalez-Cantergiani, Nicolas Hidalgo, and Erika Rosas. Disaster management platform to support real-time analytics. In *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–8. IEEE, 2016.

[30] Lisa Singh, Shweta Bansal, Leticia Bode, Ceren Budak, Guangqing Chi, Kornraphop Kawintiranon, Colton Padden, Rebecca Vanarsdall, Emily Vraga, and Yanchen Wang. A first look at covid-19 information and misinformation sharing on twitter. *arXiv preprint arXiv:2003.13907*, 2020.

[31] Ramez Kouzy, Joseph Abi Jaoude, Afif Kraitem, Molly B El Alam, Basil Karam, Elio Adib, Jabra Zarka, Cindy Traboulsi, Elie W Akl, and Khalil Baddour. Coronavirus goes viral: quantifying the covid-19 misinformation epidemic on twitter. *Cureus*, 12 (3), 2020.

[32] Catherine Ordun, Sanjay Purushotham, and Edward Raff. Exploratory analysis of covid-19 tweets using topic modeling, umap, and digraphs. *arXiv preprint arXiv:2005.03082*, 2020.

[33] Alaa Abd-Alrazaq, Dari Alhuwail, Mowafa Househ, Mounir Hamdi, and Zubair Shah. Top concerns of tweeters during the covid-19 pandemic: infoveillance study. *Journal of medical Internet research*, 22(4):e19016, 2020.

[34] Munmun De Choudhury, Michael Gamon, and Scott Counts. Happy, nervous or surprised? classification of human affective states in social media. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[35] Mohammed Jabreel and Antonio Moreno Ribas. Sitaka at semeval-2017 task 4: Sentiment analysis in twitter based on a rich set of features. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 694–699, 2017.

[36] Mohammed Jabreel and Antonio Moreno. Sentirich: Sentiment analysis of tweets based on a rich set of features. In *CCIA*, pages 137–146, 2016.

[37] Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*, 2018.

[38] Hardik Meisheri and Lipika Dey. Tcs research at semeval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299, 2018.

[39] Ji Ho Park, Peng Xu, and Pascale Fung. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and# hashtags. *arXiv preprint arXiv:1804.08280*, 2018.

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[41] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[42] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[43] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*, 2015.

[44] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[45] Akshi Kumar, Saurabh Raj Sangwan, Anshika Arora, Anand Nayyar, Mohamed Abdel-Basset, et al. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access*, 7:23319–23328, 2019.

[46] Chris Alberti, Kenton Lee, and Michael Collins. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*, 2019.

IV. A DEEP LEARNING APPROACH FOR IDEOLOGY DETECTION AND POLARIZATION ANALYSIS DURING THE COVID-19 PANDEMIC LEVERAGING SOCIAL MEDIA

Md Yasin Kabir and Sanjay Madria
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri 65401
Email: mkabir@mst.edu and madrias@mst.edu

ABSTRACT

Polarization analysis is critical for effective policy and strategy implementation. Various aspects of the COVID-19 pandemic are discussed on social media platforms extensively. While social media are used to share factual information and official directives, there is also an abundance of misinformation and beliefs (both personal and political). Some of that misinformation and beliefs are driven by polarized opinions from different ideologies. Consequently, considerable polarization has been observed on widely discussed topics related to Covid-19 such as face masks and vaccines. The study of emotion is essential for polarization detection as positive or negative sentiment towards a topic might indicate favorability or hesitancy. While positive or negative sentiment indicates a polar view toward a subject matter, it is paramount to understand the fine-grained emotion (e.g. Happiness, Sad, Anger, Pessimism) for effective polarization detection. In this research work, we propose a deep learning model leveraging the pre-trained BERT-base to detect the political ideology in the tweets for political polarization analysis. The experimental results show a considerable improvement in the accuracy of ideology detection when we use emotion as a feature. Additionally, we develop a deep learning model accompanied by an adversarial sample generation module to detect the emotion in the tweets. The adversarial sample general module significantly improves the performance of the deep learning model. Finally, we explore the political polarization for the topics "mask" and "vaccine" in the different states of the USA throughout the pandemic.

Keywords: COVID-19, Coronavirus, Polarization, Social Media, Twitter, Emotion analysis, Data Analysis, Machine Learning.

1. INTRODUCTION

COVID-19 pandemic forces people to stay at home which amplified the use of social media. The general population as well as officials leverage social media to disseminate information, directives, and to create awareness. People became engaged in social media to express their opinions, beliefs, political agenda along with different types of contents. Various topics related to Covid-19 with great interest have emerged throughout the pandemic. "Mask, Vaccine, Stay at home" are some examples of the topics with higher engagement where people with different ideologies reacted differently on those topics. People actively decide their contents of interest which often brings the same ideology people together because of the recommendation system and followers network of the social media. During the pandemic, researchers have observed a concerning amount of bias and political polarization [1, 2]. Moreover, the 2020 presidential election of the United States greatly influence people's opinions and beliefs related to Covid-19. Prior research works [3] show that such political events can reinforce people beliefs through confirmation bias. Various research works have shown that social media is highly prone to echo chambers [4]. An echo chamber is a situation where certain beliefs or assumptions (both true/false) are reinforced by repeated communication and information sharing. Most of the prior works in polarization and echo chamber detection discusses mostly political topics. However, during the pandemic researchers have found polarization and conspiracies for general health directives such as mask and vaccines [5, 6]. The use of face masks became highly polarized and a topic of debate because of different guidelines. There are various kinds of masks available and not all kinds of masks prevent COVID-19 infection at the same level. General types of masks such as surgical masks, reusable cloth masks, and face coverings do not prevent infections much while those reduce the transmission compared to professional N-95 masks. To prevent the panic buying and hoarding, some official guidelines asked people not to buy masks stating that masks are not effective [9]. After a few weeks, in April 2020, CDC urged all Americans to use a face mask. This kind of information created confusion among the general population that also instigates to polarization.

Understanding polarization is very critical to motivate the mass population effectively and create acceptable policies. A proper study of polarization for some topics can be useful for other topics that might arise in the future. It will also assist in sharing acceptable information across all demographic. While there are research works on COVID-19 polarization detection [6, 7, 8], most of those works use relatively small data sets or periods. In this work, we aim to study polarization using large-scale Twitter data collected from March 2020 to December 2021 regarding masks and vaccines. Our primary research contributions are:

- We leverage deep learning and develop a transformer-based model to detect the partisanship in the tweet. We have also created a pipeline to semi-automatically annotate the political affiliation to create the data-set for the model. Instead of using high-level sentiment (positive, neutral, and negative), we have used different emotion categories.

- To detect the emotion in a tweet, we propose a BiLSTM model with an adversarial sample generation module.

- We explore and report on polarization analysis during COVID-19 pandemic on "Mask" and "Vaccine" with respect to political ideology in the USA. We also explore the polarization in four different states during 2020 and 2021.

[9] Surgeon General Urges the Public to Stop Buying Face Masks - https://www.nytimes.com/2020/02/29/health/coronavirus-n95-face-masks.html

2. RELATED WORKS

2.1. POLARIZATION DETECTION USING TWITTER

Polarization detection within social media content especially Twitter is a popular topic of research these days. During the COVID-19 pandemic, researchers find extensive social and political polarization in the social media content. To detect the polarization and political ideology in the tweets, researchers mostly take two approaches: content-based and network-based. User metadata, tags, tweet, location, and other information are used in content-based approaches [9]. In this approach, authors use user metadata information and compared that information with seed users (verified profile with political affiliation) to infer the political ideology. While this method works well it also skews the result because of similarity in the shared content and the location of the user. In the second approach, the user network is used to detect partisanship. The network is built using retweets, engagement, and followers [7]. Ideally the more interaction someone has with seed users with specific affiliation, it is more likely for that user to follow the same political ideology. Authors in [7] created an interaction network using retweets. The authors explore the echo chambers using the interaction network. While it is true that people who interact with each other might have a similar ideology, it is also possible for people to interact with someone who oppose the content.

Topic-specific polarization exploration also became very popular during the COVID-19 pandemic. Yeung et al. [6] explore the polarization on personal face masks during the pandemic. The authors analyze the people with different demographic such as age, gender, geographic region, and household income. The authors use the valence-aware sentiment analysis to detect the polarity. The authors employ a content-based approach to detect the political ideology using a set of filtered keywords and follow networks. Jiang et al. [1] studied the polarization between different ideological groups on COVID-19 vaccine favorability and hesitancy. The authors use follower scores and expressions in the tweets to detect political affiliation. The authors examined whether and how people's opinions on the COVID-19 vaccine vary. While the above approaches can determine the political ideology those approaches are impacted by the considered sample data, sample size, and seed users. In this work, we take a content-based approach to detect political ideology. However, instead of using the profile meta-information or user network, we use the content in the tweet text and the emotion in the text to detect the political ideology. This method reduces the bias as instead of looking at user interaction network its focuses on the contents and opinions of that user.

2.2. ADVERSARIAL GENERATION FOR TEXT CLASSIFICATION

Although there are many works available on tweet classification we have found only a few attempts to classify the tweets emotion during the COVID-19 pandemic using context-based machine learning models due to the lack of available data sets. Most of the traditional tweet emotional classification works treat the problem as a text classification problem and rely on a large amount of labeled data and focus mostly on effective feature engineering. Baziotis et al. [10] and Meisheri et al. [11] who hold the first and second place of the multi-label emotion classification task of SemEval-2018 Task1, developed classifiers using a bidirectional LSTM with an attention mechanism. Using two different trained models: regularized linear regression and logistic regression classifier chain, Park et al. [12] try to classify the emotions for the same problem discussed above. The authors captured the correlation of emotion labels using logistic regression classifiers. However, none of those works perform emotion classification on a crisis dataset which might represent a wide verification of emotions with unbalanced labeled data. Yang et al. [13] introduce a COVID-19 dataset and implemented XLNet, AraBert, and ERNIE for classifying the emotion in English, Arabic, and Chinese language texts. However, the authors did not attempt any adversarial approach or any other technique to make the model better context-aware.

Adversarial learning approaches are widely popular for computer vision problems such as image classification and segmentation. However, in recent years adversarial learning gaining popularity in the field of Natural Language Processing (NLP). In [14], a team of researchers from Google and OpenAi, introduce adversarial training methods for semi-supervised text classification. The authors introduced perturbations in the text embedding in a Recurrent Neural Network (RNN) and achieved the state-of-the-art result. In a recent work, Daniel et al. [15] explore domain adversarial training for low-resource text classification. The authors claimed that transfer learning from one language to another low-resource language using adversarial technique is highly beneficial. The authors extended domain-adversarial neural network architecture to multiple source domains and evaluate the model performance to prove their claim. Authors in [16] used Generative Adversarial Learning to improve the BERT [17] and make it robust for text classification. The authors found that adopting adversarial training to enable semi-supervised learning in Transformer-based architectures improves the model performance with fewer labeled examples. While research is scarce works on adversarial learning for emotion classification from text, the idea is gaining traction recently. In December 2020, Bo Peng et al. [18] proposed an adversarial learning method for sentiment word embedding to force a generator to create word embedding with high-quality utilizing the semantic and sentiment information. In [19], the authors utilize adversarial multi-task learning for Aggressive language detection (ALD) from tweets. The authors deploy a task discriminator for text normalization to improve the ALD. The adversarial framework uses the private and shared text encoder to learn the underlying common features across the labels and thus improve the performance. In [20], the authors developed a confrontation network and used transfer learning to achieve rapid theme classification and emotion detection from the text. The authors developed an adversarial network to extract the common features of different tasks to improved the performance. In most of these works, authors are taking adversarial approaches for word embedding or transfer learning across the domains. However, due to the short text in social media contents (e.g. Tweets), the emotion can change for a single to few words where 80-90% of the words remain the same. In this work, we explore an adversarial learning approach to extract the common features across different emotions to improve emotion detection.

3. DATA PREPARATION

Figure 1, represents the basic workflow of the data collection, processing, annotation, and manipulation. We have started collecting COVID-19 related tweets originating from USA using Twitter Streaming API from March 5th, 2020. Since then we have collected over 800 million tweets until December 2021. During filtering and pre-processing steps we have discarded non-English tweets, removed the duplicates and media only tweets. Further, we process the user profile information to keep the essential information such as location, verified status, and profile description.

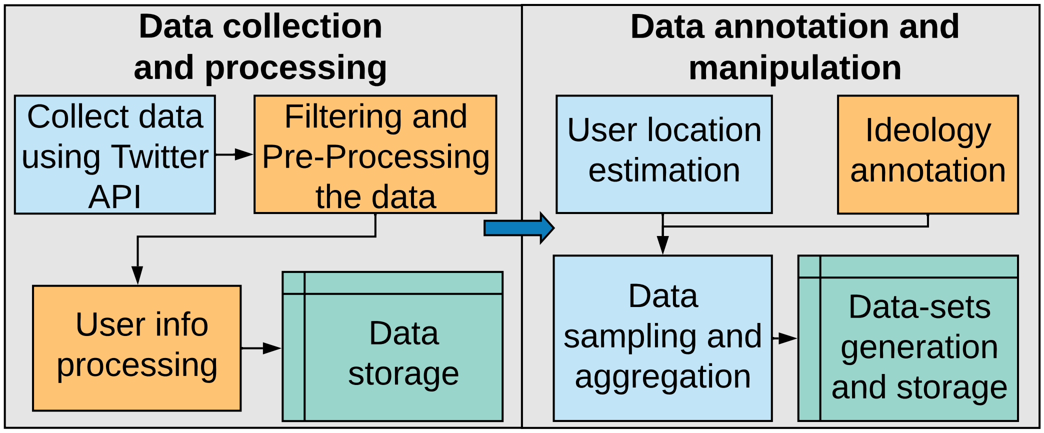

Data collection and processing
Collect data using Twitter API
Filtering and Pre-Processing the data
User info processing
Data storage
Data annotation and manipulation
User location estimation
Ideology annotation
Data sampling and aggregation
Data-sets generation and storage

Figure 1. Data processing, annotation, and manipulation

To estimate the location of the user we have used the Geo-information available with the tweets. However, less than 1% of tweets contain the geotag because of Twitter location privacy. In that case, we have used the user profile description and location meta information to estimate the location of the user. The pre-processed data is stored as CSV files. Table 1 contains the data summary information after the prepossessing steps.

Table 1. Data Summary

Attribute	**Summary**
Collection Period	Mar 5, 2020 to Dec 31, 2021.
Total Tweets	831,072,693.
Raw Data Size	11.38 Terabytes
After pre-processing	512,148,346.
Unique users	Total: 57,35,936; Verified: 59,883;

Data Annotation: For this work, we have annotated two sets of data. First, we manually annotate emotions in 10K tweets. Three annotators worked independently to annotate the emotion types in every single tweet. The detailed emotion annotation process is available in [21]. The tweets were annotated in 10 different emotion types (e.g. neutral, optimistic, happy, sad, surprise, fear, anger, denial, joking, pessimistic). To reduce the biases of the data annotation, we have added 4000 more emotion-labeled tweets publicly available at [13].

Further, we develop a semi-automatic annotation module to annotate the political ideology of a user. We have considered the members of the US Congress and self-claimed verified users with political ideology (Democratic, Republican) as the seed user. We annotated the tweets from those user profiles with the political affiliation. We only consider the original tweets by those users in the finalized data set which contains around 250K of tweets on masks and vaccines. We also identify the emotions in those tweets using the developed emotion detection model. We use this data set to train a transformer-based model and further use that model to identify the political affiliation in the other tweets.

4. EMOTION CLASSIFICATION

We develop a Deep Neural Network with adversarial sample generation and learning to classify the tweet text into a specific emotion category. The classification model comprises 6 primary components which are the input layer, embedding layer, Bidirectional Long-Short Term Memory (BiLSTM) layers, auxiliary features input, and output layers. A detailed description of each component is available in our previous research work [21]. In our prior work, the developed BiLSTM model outperforms other state of the arts in several metrics. To improve the proposed BiLSTM model for the minor emotion classes such as ’Jokes’, we introduce a method for adversarial sample generation and learning which effectively increases the performance of the BiLSTM model while converging with fewer epochs of training. Figure 2, represents the basic idea of the proposed adversarial approach to detect the emotion labels. The process can be separated into 3 steps. The first step is adopted from our previous work and described in [21]. In the following subsections, we briefly describe steps two and three.

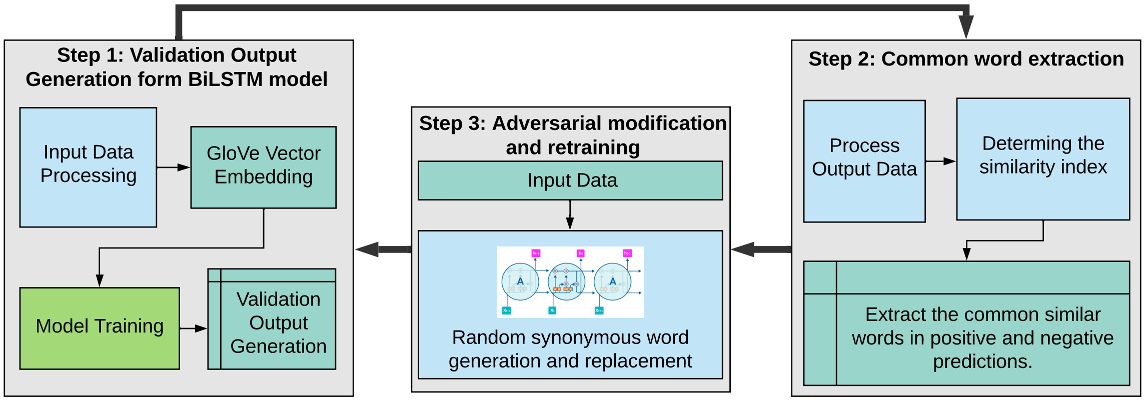

Step 1: Validation Output Generation form BiLSTM model
Input Data Processing
GloVe Vector Embedding
Model Training
Validation Output Generation
Step 3: Adversarial modification and retraining
Input Data
Random synonymous word generation and replacement
Step 2: Common word extraction
Process Output Data
Determing the similarity index
Extract the common similar words in positive and negative predictions.

Figure 2. Adversarial Training Steps

4.1. COMMON WORD EXTRACTOR

The algorithm for common word determination from the text is presented in Algorithm 1. The algorithm takes the validation prediction outputs of the model during the training and calculates the similarity ratio and distance to determine the common words. We have use Normalized hamming similarity [22] to calculate the similarity. The algorithm compared the similarity ratio and distance between the positive and negative prediction to determine which words are critical and presents across right and wrong predictions. It uses two threshold values α and δ to select the appropriate words that can be replaced to generate adversarial sample. During different training epochs the model change the values of α and δ to ensure different word selection and hence maximize the performance.

4.2. ADVERSARIAL SAMPLE GENERATION

Let us denotes the sequence of commons words across positive and negative labels as $\{W_d | d = 1, 2, 3, ..., N\}$. The module uses the pre-trained GloVe embedding vectors and seeks similar semantic words. It follows the steps in Algorithm 2. The algorithm initialize a probability value $P = 0.2$ and select a word randomly from the embedding space to replace the original word in the tweet. We have examined different probability values and found that 0.2 is the optimal one. After creating the adversarial sample this module forward those samples and combine those with the input data to create a new set of training data for the model.

Algorithm 3 Common Word Determination

```
Input: Validation Prediction $P_v$. Here [v = 1 to k]. Initial values for Similarity and Distance
    thresholds $\alpha$, $\delta$.
Output: Words dict $W_d$.
Initialization: Appends $P_v$ with input text.
for $i = 1$ to len($P_v$) do
    Calculate word frequency $W_f = [word, count]$.
    Calculate similarity index $S_{idx}$ in the predictions.
end for
Determine the similarity ratio $S_r$ and distance $S_d$ between positive preds $Pos_v$ and
    negative predes $Neg_v$.
for $i = 1$ to len($W_f$) do
    if $S_r[i] > \alpha$ and $S_d[i] < \delta$ then
        $W_d$.append($W_f.word$);
    end if
end for
return $W_d$.
```

Algorithm 4 Adversarial Sample Generation

```
Input: Sample Tweets $T$, Common words dict $W_d$.
Output: Generated sample adversarial tweets $T_{adv}$, Probability vectors $T_P$.
  Initialization: Define a initial probability P = 0.2.
  for $t$ in $T$ do
      Iterate through the common words and find out the embedding vector for each word.
      With Probability P randomly select a similar word for respective common word.
      Replace the common words with similar world in $t$.
      Append the Probability P in $T_P$.
  end for
  return $T_{adv}, T_P$.
```

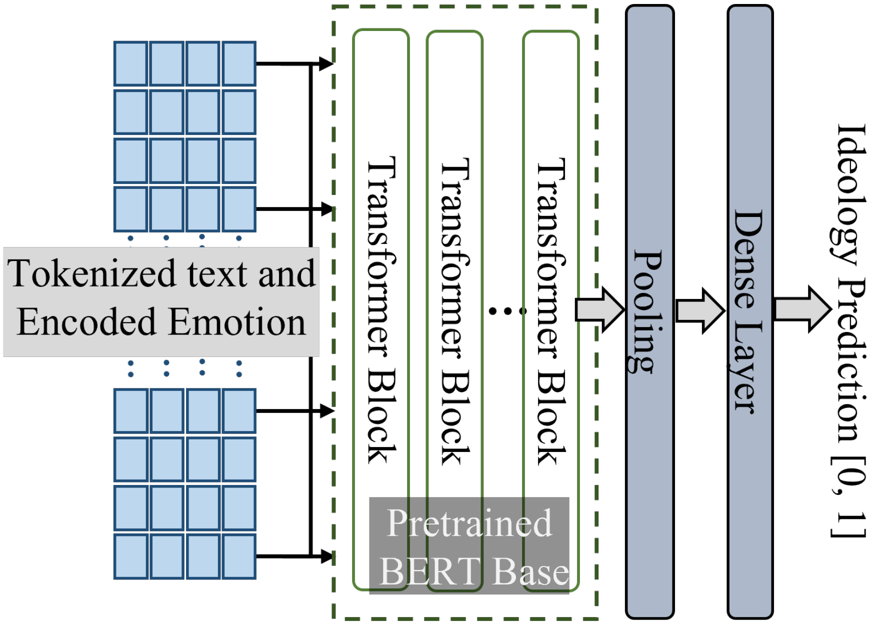


Figure 3. BERT-based transformer model for ideology detection

5. POLITICAL IDEOLOGY DETECTION

Figure 3 represents the architecture for political ideology detection using BERT-base transformer. BERT (Bidirectional Encoder Representations from Transformers) use transformers which is essentially an attention mechanism that learns contextual relations in the text. The model use encoders to learn the inputs and decoders to produce the output. In our work, we leverage a pre-trained BERT-base model and retrained it with the annotated political data-set. Along with the text, we also use the emotion class label as an input. We use a total of 150K annotated tweets for the training and validation. 80% of the data is used during training and the rest of the data were used to evaluate the model performance. During the test, we have observed that the addition of the emotion label improves the performance of the model significantly.

6. EXPERIMENTAL RESULT AND ANALYSIS

All of the experiments of this project is performed using a machine comprises of Intel® Core™ i9-9900K CPU, 64GB RAM and an Nvidia RTX-2080Ti GPU.

Table 2. Hyperparameter values for emotion classification model

Hyperparameters	Description
Text embedding	Dimension: 250
BLSTM Layer	2 layers; 250 hidden units in each
Dense Layer	2 layers; 150 and 75 units respectively
Drop-out rate	Word Embedding: 0.3; Dense layer: 0.2 each;
Activation function	ReLU; Output activation: Sigmoid;
Adam optimizer	Learning rate = 0.0001; $beta_1$=0.8;
Validation	Training and Validation Split = 80/20;
Epochs and batch	Epochs = 25; batch size = 256;

6.1. HYPERPARAMETERS OF THE MODELS

We have performed rigorous hyper-parameters tuning and found that the values in Table 2 and Table 3 are ideal to reach the desired performance. The tables also presents the layers information that are used in both models. For emotion classification we have performed parameter tuning and optimize the model after the adversarial sample learning. We used the same set of parameters as presented in the tables for performance evaluation and model re-production.We have used 80/20 training and testing split for both of the models.

Table 3. Hyperparameter values of ideology detection model

Hyperparameters	Description
Pre-trained Model	BERT Base
Linear layer	768, Activation: ReLU;
Criterion	Cross entropy loss; Optimizer: Adam
Learning rate	0.0001; Drop-out rate: 0.5
Validation	Training and Validation Split = 80/20;
Epochs and batch	Epochs = 25; batch size = 128;

6.2. EVALUATION METRICS

To evaluate the classification model, we have used F1-Micro, F1-Macro, and Accuracy as metrics. Let L denotes the number of categories, TP denotes True Positive, FP denotes False Positive, and FN denotes False Negative. We can define the F1-micro average and F1-macro average score as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

$$F1_{macro} = \frac{1}{|L|}\sum_{k=1}^{L} F1_k \tag{4}$$

$$Precision_{micro} = \frac{\sum_{k=1}^{L} TP_k}{\sum_{k=1}^{L}(TP_k + FP_k)} \tag{5}$$

$$Recall_{micro} = \frac{\sum_{k=1}^{L} TP_k}{\sum_{k=1}^{L}(TP_k + FN_k)} \tag{6}$$

$$F1_{micro} = \frac{2 * Precision_{micro} * Recall_{micro}}{Precision_{micro} + Recall_{micro}} \tag{7}$$

$$Accuracy = \frac{1}{T}\sum_{k=1}^{T} \sigma(Y_k == P_k) \tag{8}$$

Accuracy is used as a metric for model performance as it can give a better observation for imbalanced categories. Equation 8 defines the Accuracy score where $\sigma(Y_k == P_k)$ returns 1 if the prediction is correct, otherwise 0. To evaluate the performance of the political ideology detection model we have used the accuracy score and ROC AUC(Area under the ROC Curve). ROC curve shows the performance of a classification model at all the given classification thresholds. We have used "sklearn" metrics package to calculate the ROC AUC score.

6.3. EXPERIMENTAL RESULTS

Table 4. Classifier Evaluation and Comparison

Models	**F1-Micro**	**F1-Macro**	**Accuracy**
SVM-Unigrams	0.53	0.41	0.74
NTUA-SLP	0.60	0.49	0.85
BiLSTM$_{Aux}$	0.55	0.54	0.86
BiLSTM$_{Aux}$+ADV	0.64	0.61	0.91

Table 4, represents the primary experimental results. To test the adversarial approach, we have integrated the module with the above described BiLSTM network with auxiliary feature engineering. From the table, we can observe that the adversarial integration improved the performance considerably. The F1-Micro and F1-Macro scores are comparatively lower as we have some emotion categories with a very small number of labels (e.g denial, joking, pessimistic). All of the models perform poorly for those classes. The performance evaluation for the different models for political ideology detection is present in Table 5. We can see that the BERT with the addition of the emotion category outperforms other models. We have also evaluated RoBERTa with and without the the emotion labels. The performance of RoBERTa model is also very close to BERT models.

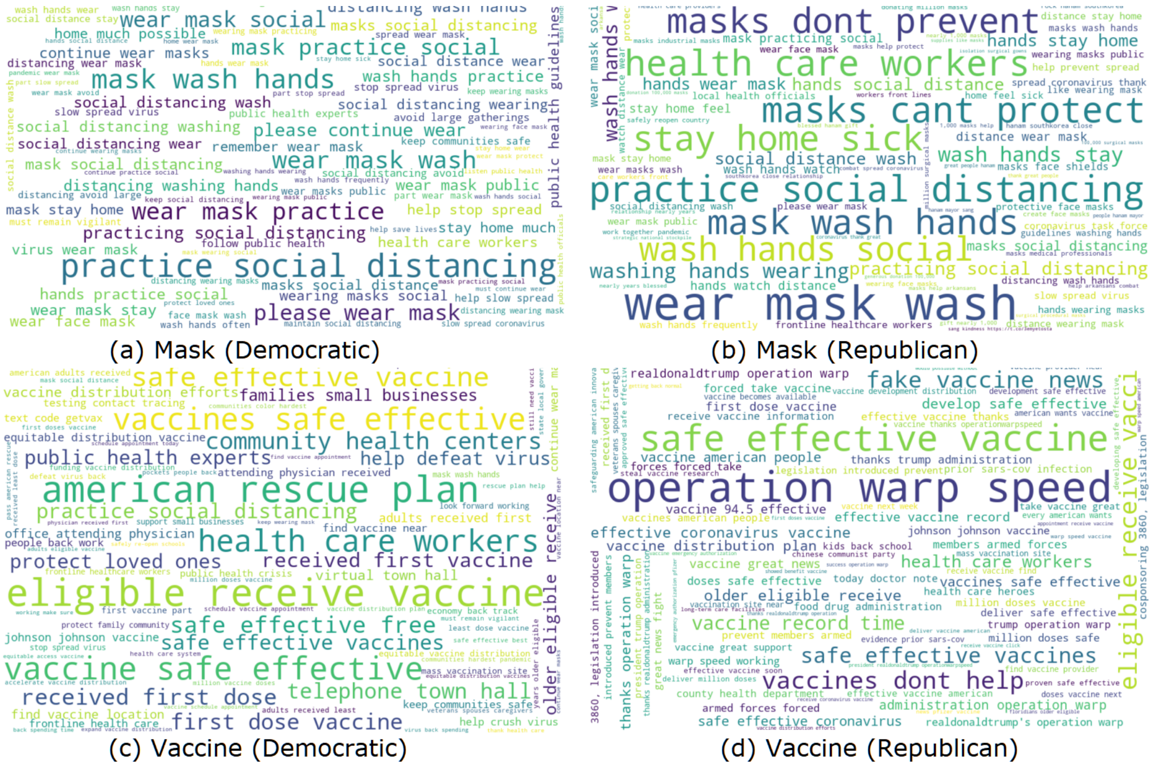
Table 5. Model performance evaluation for ideology detection

Models	**Accuracy**	**ROC AUC**
SVM	0.73	0.70
CNN	0.79	0.76
RoBERTa	0.84	0.83
BERT	0.85	0.83
BERT$_{emot}$	0.88	0.87

6.4. COVID-19 POLARIZATION ANALYSIS

The primary aim of the polarization analysis is to find out the polarized topics and explore the change in sentiment during the pandemic. Figure 4 depicts the word-clouds on the topics masks and vaccines using trigrams from the tweets by people with democratic and republican ideology. For each word cloud, a sample of 1 million tweets was used. We observe some polarized opinions across the demographics. For instance, trigrams "masks don't prevent" and "masks cant prevent" are prominent in republication tweets. Similarly, we observe a different set of discussions on vaccines. In the republican tweets "operation wrap speed" and "fake vaccine news" appears frequently compared to democratic tweets. Figure 5(a, b) represents the monthly polarity on the topics "Masks" and "Vaccines" in the

USA. In the charts the blue lines represents the polarization score for democratic ideology and the red lines represents republican ideology score.



(a) Mask (Democratic)

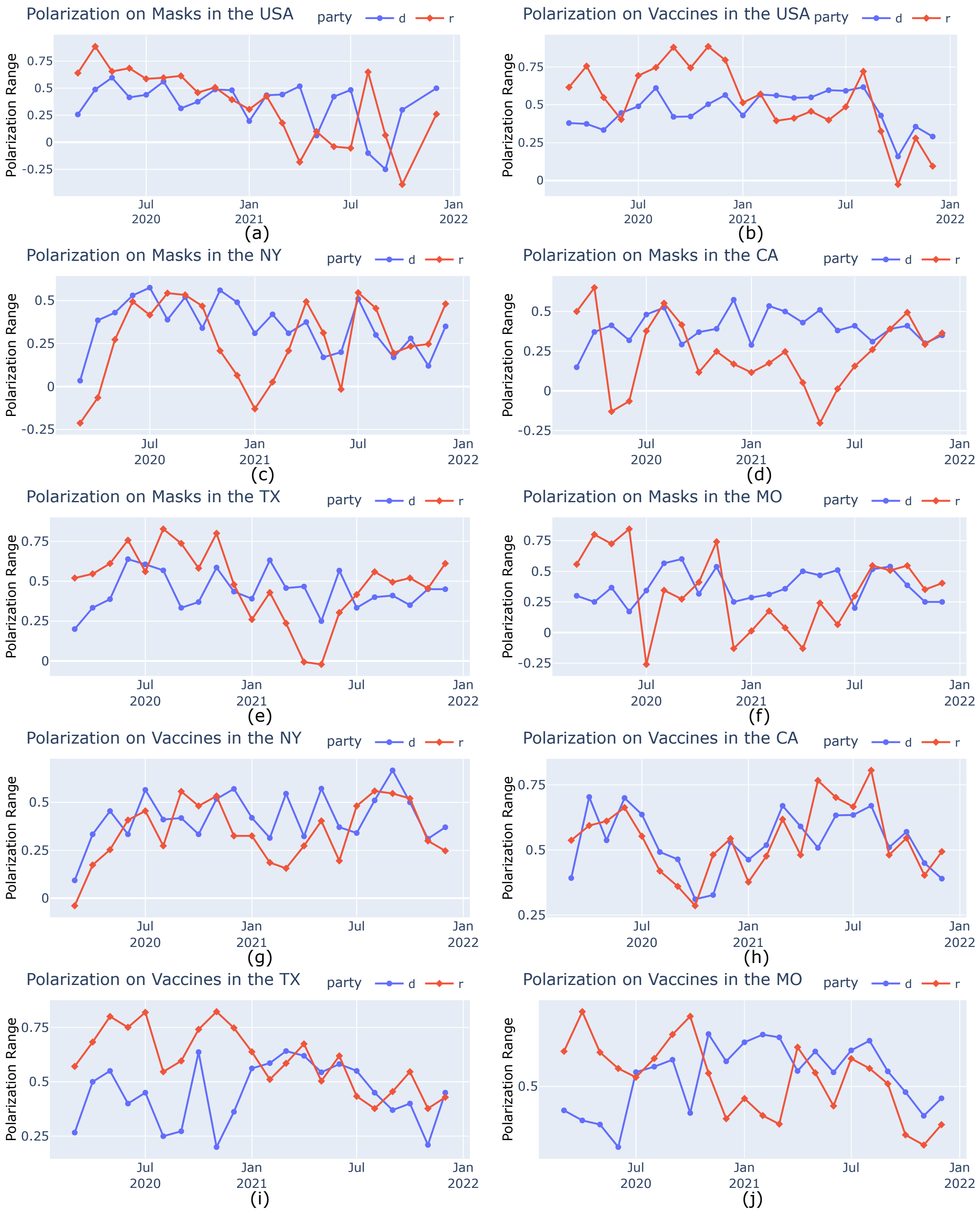
(b) Mask (Republican)

(c) Vaccine (Democratic)

(d) Vaccine (Republican)

Figure 4. Discussion on masks and vaccines since March 2020 to December 2021

To calculate the polarity values one million tweets per month is selected from each ideology class. The polarity for each tweet is set as positive one (+1) for the positive emotion and negative one (-1) for the negative emotion. Further, we take the average of the polarity in all the tweets to calculate the polarization score. Therefore, the lower polarity score denotes the negative sentiment on the topics compared to the opposite ideology. We further explore the polarization in the four different states (e.g. NY - New York, CA - California, TX - Texas, and MO - Missouri) of the USA for the topic "Masks" and "Vaccines". In Figure 5 we observe some interesting contrast in the polarity for each ideology before and after the 2020 presidential election for both of the topics. The score for the republican ideology is significantly went down after the election in the Texas and Missouri while the score was much higher before the election compared to the democratic ideology.



Polarization on Masks in the USA

(a)

Polarization on Vaccines in the USA

(b)

Polarization on Masks in the NY

(c)

Polarization on Masks in the CA

(d)

Polarization on Masks in the TX

(e)

Polarization on Masks in the MO

(f)

Polarization on Vaccines in the NY

(g)

Polarization on Vaccines in the CA

(h)

Polarization on Vaccines in the TX

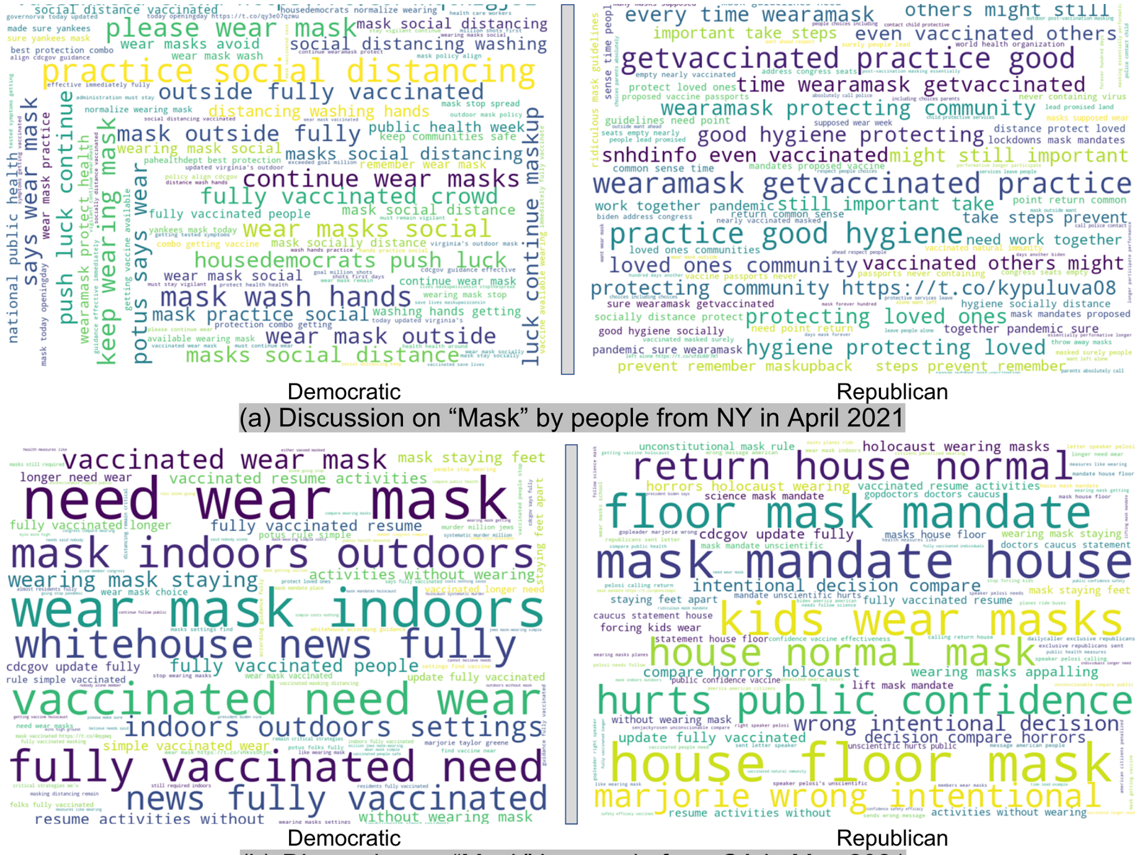
(i)

Polarization on Vaccines in the MO

(j)

Figure 5. Polarization on Masks and Vaccines over in 2020 and 2021 in the USA

Figure 5(c) to 5(j) represents the polarization on "masks" and "vaccines" in the four states. While all of the four states have the similar trend we can see some distinct differences. For instances in Figure 5(c), in April 2021 for the New York state we see a higher polarity score for republican ideology compared to other three states. In the other states the democratic polarity score on topic "Masks" was much higher compared to the republication. We can see another interesting observation in Figure 5(d) for the month May 2021 in California state. We can see that the republican sentiment went down significantly in May 2021.

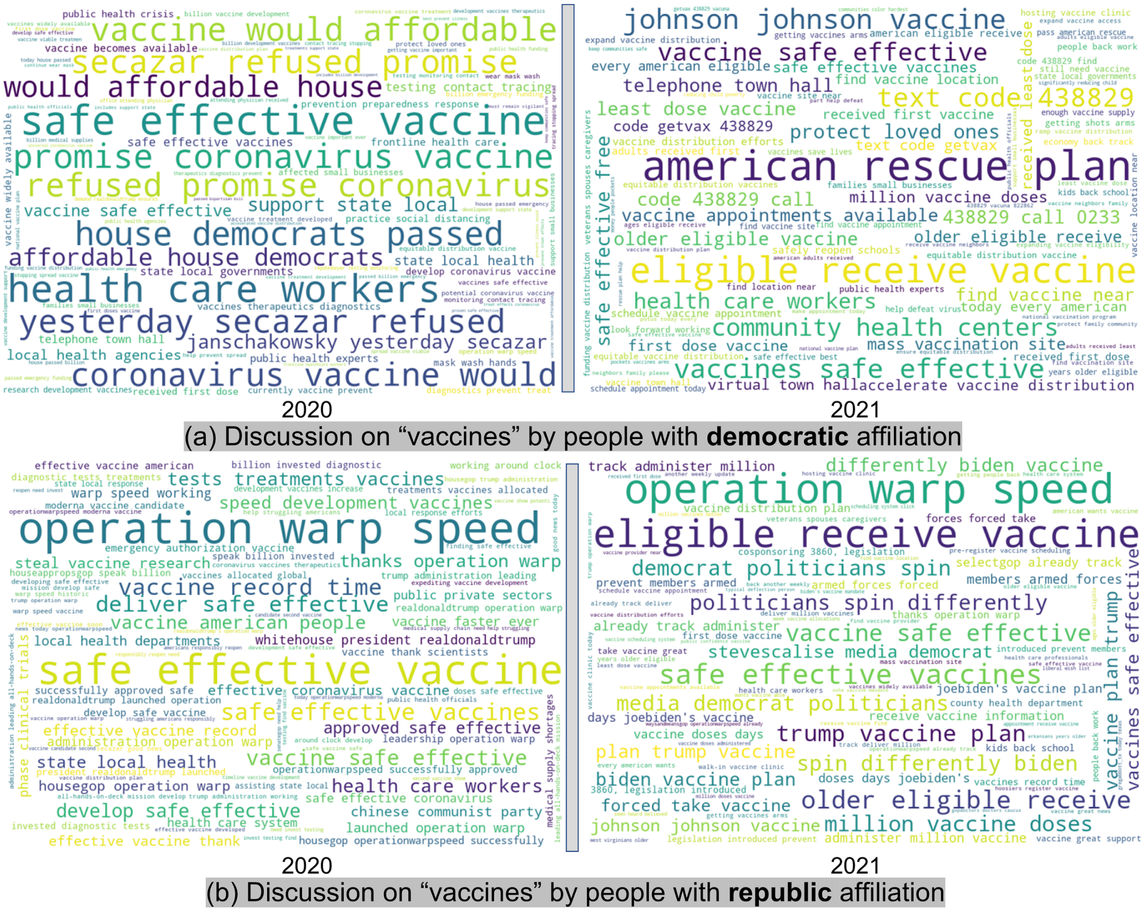


(a) Discussion on “Mask” by people from NY in April 2021

(b) Discussion on “Mask” by people from CA in May 2021

Figure 6. Discussion on Mask by people from New York and California

We present the topics of interest during the above mentioned events in Figure 6. We observe the people in NY have considerably similar discussion and emotion on the masks in April 2021. Although there was a shift in the sentiment after the 2020 election, the people of NY decided to work together to contain the corona virus. We observe a very different set of discussions and emotions among the democratic and republican in the California. While the democrats want to enforce the use of mask, many republican stand strongly against it. We can see some trigrams such as ’hurts public confidence’, ’forcing kids wear’, and ’lift mask mandate’ in the tweets with republican ideology.



(a) Discussion on “vaccines” by people with **democratic** affiliation

(b) Discussion on “vaccines” by people with **republic** affiliation

Figure 7. Discussion on vaccines during 2020 and 2021

In Figure 7 we can observe the contrast between the discussed topics related to "vaccines" by democrats and republicans during 2020 and 2021. People with democratic ideology were strong vocal for the vaccine development and affordable vaccines during 2020. They felt the initiatives taken by the authorities is not enough. On the contrary republicans were praising about operation wrap speed and safe effective vaccine. In 2021 democrats discussed more about American rescue plan and mass vaccination with the aim of encourage people to take vaccine. However, the republican are vocal against the political spinning of vaccine success and also vocal against forced vaccination policy by different authorities and employers. Due to the page limitation we present limited number of analysis in the paper. However, we plan to make more analysis public along with our data and finding in the github repository [10]. The repository will contains the interactive graphs which will be accessible through a github website.

7. CONCLUSION AND FUTURE WORK

In this work, we have extend our emotion detection model and achieve higher scores compare to the state-of-the-art models for emotion classification. We also propose a BERT-based political ideology classification model where we use emotions as a feature for polarization detection using the tweet text only. The performance evaluation shows that the proposed model outperformed other well-known models for the classification task. Further, we have to use the classified ideology label to analyze the tweets and explore the polarization on the topic "mask" and "vaccine" during the COVID-19 pandemic. In the future, we plan aim to study the influence of the bias in our study and conduct experiments with different sample sizes for different periods. We also aim to perform rigorous polarization analysis and share those analyses in public using the GitHub repository.

REFERENCES

[1] Xiaoya Jiang, Min-Hsin Su, Juwon Hwang, Ruixue Lian, Markus Brauer, Sunghak Kim, and Dhavan Shah. Polarization over vaccination: Ideological differences in twitter expression about covid-19 vaccine favorability and specific hesitancy concerns. *Social Media+ Society*, 7(3):20563051211048413, 2021.

[10] https://github.com/mykabir/Covid-19-Polarization-Analysis

[2] Julie Jiang, Emily Chen, Shen Yan, Kristina Lerman, and Emilio Ferrara. Political polarization drives online conversations about covid-19 in the united states. *Human Behavior and Emerging Technologies*, 2(3):200–211, 2020.

[3] Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10):1531–1542, 2015.

[4] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of communication*, 64(2):317–332, 2014.

[5] Emily Chen, Herbert Chang, Ashwin Rao, Kristina Lerman, Geoffrey Cowan, and Emilio Ferrara. Covid-19 misinformation and the 2020 us presidential election. *The Harvard Kennedy School Misinformation Review*, 2021.

[6] Neil Yeung, Jonathan Lai, and Jiebo Luo. Face off: Polarized public opinions on personal face mask usage during the covid-19 pandemic. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4802–4810. IEEE, 2020.

[7] Julie Jiang, Xiang Ren, Emilio Ferrara, et al. Social media polarization and echo chambers in the context of covid-19: Case study. *JMIRx med*, 2(3):e29570, 2021.

[8] Jon Green, Jared Edgerton, Daniel Naftel, Kelsey Shoub, and Skyler J Cranmer. Elusive consensus: Polarization in elite communication on the covid-19 pandemic. *Science advances*, 6(28):eabc2717, 2020.

[9] Aseel Addawood, Adam Badawy, Kristina Lerman, and Emilio Ferrara. Linguistic cues to deception: Identifying political trolls on social media. In *Proceedings of the international AAAI conference on web and social media*, volume 13, pages 15–25, 2019.

[10] Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*, 2018.

[11] Hardik Meisheri and Lipika Dey. Tcs research at semeval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299, 2018.

[12] Ji Ho Park, Peng Xu, and Pascale Fung. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and# hashtags. *arXiv preprint arXiv:1804.08280*, 2018.

[13] Qiang Yang, Hind Alamro, Somayah Albaradei, Adil Salhi, Xiaoting Lv, Changsheng Ma, Manal Alshehri, Inji Jaber, Faroug Tifratene, Wei Wang, et al. Senwave: Monitoring the global sentiments under the covid-19 pandemic. *arXiv preprint arXiv:2006.10842*, 2020.

[14] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[15] Daniel Grießhaber, Ngoc Thang Vu, and Johannes Maucher. Low-resource text classification using domain-adversarial learning. *Computer Speech & Language*, 62: 101056, 2020.

[16] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, 2020.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Bo Peng, Jin Wang, and Xuejie Zhang. Adversarial learning of sentiment word representations for sentiment analysis. *Information Sciences*, 541:426–441, 2020.

[19] Shengqiong Wu, Hao Fei, and Donghong Ji. Aggressive language detection with joint text normalization via adversarial multi-task learning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 683–696. Springer, 2020.

[20] E Haihong, Hu Yingxi, Peng Haipeng, Zhao Wen, Xiao Siqi, and Niu Peiqing. Theme and sentiment analysis model of public opinion dissemination based on generative adversarial network. *Chaos, Solitons & Fractals*, 121:160–167, 2019.

[21] Md Yasin Kabir and Sanjay Madria. Emocov: Machine learning for emotion detection, analysis and visualization using covid-19 tweets. *Online Social Networks and Media*, 23:100135, 2021.

[22] P Rajarajeswari and N Uma. Normalized hamming similarity measure for intuitionistic fuzzy multi sets and its application in medical diagnosis. *International Journal of Mathematics Trends and Technology*, 5(3):219–225, 2014.

SECTION

3. CONCLUSION AND FUTURE WORK

This section discusses the major contributions and future directions of this research work.

3.1. PRIMARY CONTRIBUTIONS

The primary aim of this research work is to develop efficient applications to process social media data and extract valuable information for effective disaster management. Extracted information and insights from Twitter are useful at the different stages of a disaster. The goal of this work includes Twitter data analysis for ongoing disasters as well as preventing future disastrous events. To achieve those goals this work proposes several methods and tools for real-time social media data collection and analysis. During this research work, an application is developed to fetch and process tweets in real-time leveraging GPU processing. This work proposes novel deep learning pipelines for tweet classification in order to detect the people who need assistance during a disaster. This work also proposes and demonstrates an effective rescue scheduling method to assist the stranded people during an ongoing crisis.

Sentiments at the personal and community level play a crucial role during an event. Depending on the different characteristics it can provide valuable insights and indications about the ongoing and possible future events (e.g. protest, hate crime, polarization, election results, etc.). To understand the sentiment in the tweets this work proposes a novel method for emotion detection using the tweet text. This research presents EMOCOV which leverages the deep neural network to understand the context in the text and identify the emotion type expressed in the tweet. Further, this work proposes an adversarial sample generation and learning mechanism to improve the emotion classification. The adversarial module replaces the common words in the text from a given vector space and generates new training samples. The performance evaluation shows considerable improvement in emotion classification after the addition of adversarial sample generation. A demonstration of the emotion classification and evolution of different emotions during the COVID-19 pandemic is also included in this research.

During the COVID-19 pandemic, a concerning amount of polarization, racial comments, and hate crimes has been observed on social media. Polarization detection is critical to understanding the opinions and reactions of people from different demographic. While emotion analysis is paramount for polarization analysis, researchers found that polarization is highly correlated with the political views of the people. Hence, it is essential to understand the political bias for polarization analysis. This research work presents a unique transformer-based machine learning model to identify the political ideology using the tweets. The proposed classifier utilizes the expressed emotion in the tweet text to identify the political affiliation. The expressed emotion is extracted using the proposed emotion detection model with adversarial sample generation. To evaluate the performance and effectiveness of the political ideology detection this research performs a historical polarization analysis during the COVID-19 pandemic in the United States.

3.2. FUTURE RESEARCH DIRECTION

This research presents a method for political ideology detection and conducts a study on polarization during the COVID-19 pandemic in the USA and several individual states. However, there remains great scope for further analysis of the polarization and its impact on the various topic during the COVID-19 pandemic. Moreover, because of the nature of data sampling polarization studies suffer from unwanted bias. No study on the impact of the sample size and possible bias has been conducted in this study. As a future direction, I will conduct experiments to study the influence of the bias and quantify the bias for different sample sizes.

Effective and efficient topic modeling is very important to detect future events. Due to the nature of tweets, it is very challenging to identify the topics and track the evolution of those topics over time. Furthermore, it is also necessary to detect toxic comments in order to track hate words. Polarization along with hate speech detection can provide the necessary information to prevent an undesirable event. The future direction of this research work is to continue working on polarization and political bias with the aim of subtopics and hate speech detection. I would like to conduct experiments on source detection for the hate speech along with political echo chamber detection and analysis.

While GPU process the data remarkably fast, there are still many limitations. There is a scarcity of the available methods and library packages for GPU processing. During this work several GPU-based data processing methods were implemented. In the future, I aim to implement more GPU-based method, quantify the performance, and make those methods publicly available.

REFERENCES

[1] Alaa Abd-Alrazaq, Dari Alhuwail, Mowafa Househ, Mounir Hamdi, and Zubair Shah. Top concerns of tweeters during the covid-19 pandemic: infoveillance study. *Journal of medical Internet research*, 22(4):e19016, 2020.

[2] Aseel Addawood, Adam Badawy, Kristina Lerman, and Emilio Ferrara. Linguistic cues to deception: Identifying political trolls on social media. In *Proceedings of the international AAAI conference on web and social media*, volume 13, pages 15–25, 2019.

[3] Chris Alberti, Kenton Lee, and Michael Collins. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*, 2019.

[4] David E Alexander. Social media in disaster risk reduction and crisis management. *Science and engineering ethics*, 20(3):717–733, 2014.

[5] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

[6] Thayer Alshaabi, David Rushing Dewhurst, Joshua R Minot, Michael V Arnold, Jane L Adams, Christopher M Danforth, and Peter Sheridan Dodds. The growing amplification of social media: measuring temporal and social contagion dynamics for over 150 languages on twitter for 2009–2020. *EPJ data science*, 10(1):1–28, 2021.

[7] Ansari Fatima Anees, Arsalaan Shaikh, Arbaz Shaikh, and Sufiyan Shaikh. Survey paper on sentiment analysis: Techniques and challenges. *EasyChair2516-2314*, 2020.

[8] Drake Baer. As sandy became# sandy, emergency services got social. *Fast Company*, 9, 2012.

[9] Juan M Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, and Gerardo Chowell. A large-scale covid-19 twitter chatter dataset for open scientific research–an international collaboration. *arXiv preprint arXiv:2004.03688*, 2020.

[10] Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. Tweeting from left to right: Is online political communication more than an echo chamber? *Psychological science*, 26(10):1531–1542, 2015.

[11] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Coling 2010: Posters*, pages 36–44, 2010.

[12] Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*, 2018.

[13] Cody Buntain, Jennifer Golbeck, Brooke Liu, and Gary LaFree. Evaluating public response to the boston marathon bombing and other acts of terrorism through twitter. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[14] Emily Chen, Kristina Lerman, and Emilio Ferrara. Covid-19: The first public coronavirus twitter dataset. *arXiv preprint arXiv:2003.07372*, 2020.

[15] Emily Chen, Herbert Chang, Ashwin Rao, Kristina Lerman, Geoffrey Cowan, and Emilio Ferrara. Covid-19 misinformation and the 2020 us presidential election. *The Harvard Kennedy School Misinformation Review*, 2021.

[16] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[17] Xinyu Chen, Youngwoon Cho, and Suk Young Jang. Crime prediction using twitter sentiment and weather. In *2015 Systems and Information Engineering Design Symposium*, pages 63–68. IEEE, 2015.

[18] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of communication*, 64(2):317–332, 2014.

[19] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[20] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, 2020.

[21] Munmun De Choudhury, Michael Gamon, and Scott Counts. Happy, nervous or surprised? classification of human affective states in social media. In *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[23] Mark Dredze, David A Broniatowski, and Karen M Hilyard. Zika vaccine misconceptions: A social media analysis. *Vaccine*, 34(30):3441, 2016.

[24] Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*, pages 363–370, 2005.

[25] Huiji Gao, Geoffrey Barbier, and Rebecca Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 26(3):10–14, 2011.

[26] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):1–27, 2018.

[27] Matthew S Gerber. Predicting crime using twitter and kernel density estimation. *Decision Support Systems*, 61:115–125, 2014.

[28] Sujatha Das Gollapalli, Polina Rozenshtein, and See Kiong Ng. Ester: Combining word co-occurrences and word associations for unsupervised emotion detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1043–1056, 2020.

[29] Santiago González-Carvajal and Eduardo C Garrido-Merchán. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020.

[30] Francisco Jose Grajales III, Samuel Sheps, Kendall Ho, Helen Novak-Lauscher, and Gunther Eysenbach. Social media: a review and tutorial of applications in medicine and health care. *Journal of medical Internet research*, 16(2):e13, 2014.

[31] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[32] Jon Green, Jared Edgerton, Daniel Naftel, Kelsey Shoub, and Skyler J Cranmer. Elusive consensus: Polarization in elite communication on the covid-19 pandemic. *Science advances*, 6(28):eabc2717, 2020.

[33] Daniel Grießhaber, Ngoc Thang Vu, and Johannes Maucher. Low-resource text classification using domain-adversarial learning. *Computer Speech & Language*, 62: 101056, 2020.

[34] Purva Grover, Arpan Kumar Kar, Yogesh K Dwivedi, and Marijn Janssen. Polarization and acculturation in us election 2016 outcomes–can twitter analytics predict changes in voting preferences. *Technological Forecasting and Social Change*, 145: 438–460, 2019.

[35] Xinyi Guo and Jinfeng Li. A novel twitter sentiment analysis model with baseline correlation for financial market prediction with improved efficiency. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 472–477. IEEE, 2019.

[36] E Haihong, Hu Yingxi, Peng Haipeng, Zhao Wen, Xiao Siqi, and Niu Peiqing. Theme and sentiment analysis model of public opinion dissemination based on generative adversarial network. *Chaos, Solitons & Fractals*, 121:160–167, 2019.

[37] Eui-Hong Sam Han, George Karypis, and Vipin Kumar. Text categorization using weight adjusted k-nearest neighbor classification. In *Pacific-asia conference on knowledge discovery and data mining*, pages 53–65. Springer, 2001.

[38] Maruf Hassan, Md Sakib Bin Alam, and Tanveer Ahsan. Emotion detection from text using skip-thought vectors. In *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pages 501–506. IEEE, 2018.

[39] E Hirata, MA Giannotti, APC Larocca, and JA Quintanilha. Flooding and inundation collaborative mapping–use of the crowdmap/ushahidi platform in the city of sao paulo, brazil. *Journal of Flood Risk Management*, 11:S98–S109, 2018.

[40] Ali Shariq Imran, Sher Muhammad Daudpota, Zenun Kastrati, and Rakhi Batra. Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on covid-19 related tweets. *IEEE Access*, 8:181074–181090, 2020.

[41] Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. Extracting information nuggets from disaster-related messages in social media. In *Iscram*, 2013.

[42] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. *arXiv preprint arXiv:1605.05894*, 2016.

[43] Mohammed Jabreel and Antonio Moreno. Sentirich: Sentiment analysis of tweets based on a rich set of features. In *CCIA*, pages 137–146, 2016.

[44] Mohammed Jabreel and Antonio Moreno Ribas. Sitaka at semeval-2017 task 4: Sentiment analysis in twitter based on a rich set of features. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 694–699, 2017.

[45] Julie Jiang, Emily Chen, Shen Yan, Kristina Lerman, and Emilio Ferrara. Political polarization drives online conversations about covid-19 in the united states. *Human Behavior and Emerging Technologies*, 2(3):200–211, 2020.

[46] Julie Jiang, Xiang Ren, Emilio Ferrara, et al. Social media polarization and echo chambers in the context of covid-19: Case study. *JMIRx med*, 2(3):e29570, 2021.

[47] Xiaoya Jiang, Min-Hsin Su, Juwon Hwang, Ruixue Lian, Markus Brauer, Sunghak Kim, and Dhavan Shah. Polarization over vaccination: Ideological differences in twitter expression about covid-19 vaccine favorability and specific hesitancy concerns. *Social Media+ Society*, 7(3):20563051211048413, 2021.

[48] Fang Jin, Wei Wang, Prithwish Chakraborty, Nathan Self, Feng Chen, and Naren Ramakrishnan. Tracking multiple social media for stock market event prediction. In *Industrial conference on data mining*, pages 16–30. Springer, 2017.

[49] M Yasin Kabir, Sergey Gruzdev, and Sanjay Madria. Stimulate: A system for real-time information acquisition and learning for disaster management. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 186–193. IEEE, 2020.

[50] Md Kabir, Sanjay Madria, et al. A deep learning approach for tweet classification and rescue scheduling for effective disaster management. *arXiv preprint arXiv:1908.01456*, 2019.

[51] Md Kabir, Sanjay Madria, et al. Coronavis: A real-time covid-19 tweets analyzer. *arXiv preprint arXiv:2004.13932*, 2020.

[52] Md Yasin Kabir and Sanjay Madria. Emocov: Machine learning for emotion detection, analysis and visualization using covid-19 tweets. *Online Social Networks and Media*, 23:100135, 2021.

[53] Mark E Keim and Eric Noji. Emergent use of social media: a new age of opportunity for disaster resilience. *American journal of disaster medicine*, 6(1):47–54, 2011.

[54] Larry J King. Social media use during natural disasters: An analysis of social media usage during hurricanes harvey and irma. 2018.

[55] Ramez Kouzy, Joseph Abi Jaoude, Afif Kraitem, Molly B El Alam, Basil Karam, Elio Adib, Jabra Zarka, Cindy Traboulsi, Elie W Akl, and Khalil Baddour. Coronavirus goes viral: quantifying the covid-19 misinformation epidemic on twitter. *Cureus*, 12 (3), 2020.

[56] Akshi Kumar, Saurabh Raj Sangwan, Anshika Arora, Anand Nayyar, Mohamed Abdel-Basset, et al. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. *IEEE Access*, 7:23319–23328, 2019.

[57] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[58] Mark Latonero and Irina Shklovski. Emergency management, twitter, and social media evangelism. *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, 3(4):1–16, 2011.

[59] Xiangsheng Li, Jianhui Pang, Biyun Mo, and Yanghui Rao. Hybrid neural networks for social emotion detection over short text. In *2016 International joint conference on neural networks (IJCNN)*, pages 537–544. IEEE, 2016.

[60] Bruce R Lindsay. Social media and disasters: Current uses, future options, and policy considerations, 2011.

[61] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[62] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[63] Pablo Martí, Leticia Serrano-Estrada, and Almudena Nolasco-Cirugeda. Social media data: Challenges, opportunities and limitations in urban studies. *Computers, Environment and Urban Systems*, 74:161–174, 2019.

[64] Hardik Meisheri and Lipika Dey. Tcs research at semeval-2018 task 1: Learning robust representations using multi-attention architecture. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 291–299, 2018.

[65] Panagiotis Takas Metaxas, Samantha Finn, and Eni Mustafaraj. Using twittertrails. com to investigate rumor propagation. In *Proceedings of the 18th ACM Conference Companion on Computer Supported Cooperative Work & Social Computing*, pages 69–72, 2015.

[66] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[67] Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17, 2018.

[68] Ruchit Nagar, Qingyu Yuan, Clark C Freifeld, Mauricio Santillana, Aaron Nojima, Rumi Chunara, and John S Brownstein. A case study of the new york city 2012-2013 influenza season with daily geocoded twitter data from temporal and spatiotemporal perspectives. *Journal of medical Internet research*, 16(10):e236, 2014.

[69] Nuno Oliveira, Paulo Cortez, and Nelson Areal. The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with applications*, 73:125–144, 2017.

[70] Ahmed Husseini Orabi, Prasadith Buddhitha, Mahmoud Husseini Orabi, and Diana Inkpen. Deep learning for depression detection of twitter users. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 88–97, 2018.

[71] Catherine Ordun, Sanjay Purushotham, and Edward Raff. Exploratory analysis of covid-19 tweets using topic modeling, umap, and digraphs. *arXiv preprint arXiv:2005.03082*, 2020.

[72] Sunday Oluwafemi Oyeyemi, Elia Gabarron, and Rolf Wynn. Ebola, twitter, and misinformation: a dangerous combination? *Bmj*, 349:g6178, 2014.

[73] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[74] Ji Ho Park, Peng Xu, and Pascale Fung. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and# hashtags. *arXiv preprint arXiv:1804.08280*, 2018.

[75] Mina Park, Yao Sun, and Margaret L McLaughlin. Social media propagation of content promoting risky health behavior. *Cyberpsychology, Behavior, and Social Networking*, 20(5):278–285, 2017.

[76] Bo Peng, Jin Wang, and Xuejie Zhang. Adversarial learning of sentiment word representations for sentiment analysis. *Information Sciences*, 541:426–441, 2020.

[77] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[78] Jürgen Pfeffer, Thomas Zorbach, and Kathleen M Carley. Understanding online firestorms: Negative word-of-mouth dynamics in social media networks. *Journal of Marketing Communications*, 20(1-2):117–128, 2014.

[79] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[80] Andrew G Reece, Andrew J Reagan, Katharina LM Lix, Peter Sheridan Dodds, Christopher M Danforth, and Ellen J Langer. Forecasting the onset and course of mental illness with twitter data. *Scientific reports*, 7(1):1–11, 2017.

[81] Megha Sharma, Kapil Yadav, Nitika Yadav, and Keith C Ferdinand. Zika virus pandemic—analysis of facebook as a social media health information platform. *American journal of infection control*, 45(3):301–302, 2017.

[82] Irina Shklovski, Moira Burke, Sara Kiesler, and Robert Kraut. Technology adoption and use in the aftermath of hurricane katrina in new orleans. *American Behavioral Scientist*, 53(8):1228–1246, 2010.

[83] Lisa Singh, Shweta Bansal, Leticia Bode, Ceren Budak, Guangqing Chi, Kornraphop Kawintiranon, Colton Padden, Rebecca Vanarsdall, Emily Vraga, and Yanchen Wang. A first look at covid-19 information and misinformation sharing on twitter. *arXiv preprint arXiv:2003.13907*, 2020.

[84] Brian G Southwell, Jeff Niederdeppe, Joseph N Cappella, Anna Gaysynsky, Dannielle E Kelley, April Oh, Emily B Peterson, and Wen-Ying Sylvia Chou. Misinformation as a misunderstood challenge to public health. *American journal of preventive medicine*, 57(2):282–285, 2019.

[85] Patric R Spence, Kenneth A Lachlan, and Adam M Rainear. Social media and crisis research: Data collection and directions. *Computers in Human Behavior*, 54: 667–672, 2016.

[86] Kate Starbird and Leysia Palen. Pass it on?: Retweeting in mass emergency. In *ISCRAM*, 2010.

[87] Yahya M Tashtoush and Dana Abed Al Aziz Orabi. Tweets emotion prediction by using fuzzy logic system. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 83–90. IEEE, 2019.

[88] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*, 2015.

[89] Xingyou Wang, Weijie Jiang, and Zhiyong Luo. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2428–2437, 2016.

[90] Zheye Wang, Nina SN Lam, Nick Obradovich, and Xinyue Ye. Are vulnerable communities digitally left behind in social responses to natural disasters? an evidence from hurricane sandy with twitter data. *Applied geography*, 108:1–8, 2019.

[91] Nathan G Watkins, Nigel H Lovell, and Mark E Larsen. Smct-an innovative tool for mental health analysis of twitter data. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4114–4117. IEEE, 2018.

[92] Daniel Wladdimiro, Pablo Gonzalez-Cantergiani, Nicolas Hidalgo, and Erika Rosas. Disaster management platform to support real-time analytics. In *2016 3rd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, pages 1–8. IEEE, 2016.

[93] Shengqiong Wu, Hao Fei, and Donghong Ji. Aggressive language detection with joint text normalization via adversarial multi-task learning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 683–696. Springer, 2020.

[94] Jia Xue, Junxiang Chen, Chen Chen, ChengDa Zheng, and Tingshao Zhu. Machine learning on big data from twitter to understand public reactions to covid-19. *arXiv preprint arXiv:2005.08817*, 2020.

[95] Qiang Yang, Hind Alamro, Somayah Albaradei, Adil Salhi, Xiaoting Lv, Changsheng Ma, Manal Alshehri, Inji Jaber, Faroug Tifratene, Wei Wang, et al. Senwave: Monitoring the global sentiments under the covid-19 pandemic. *arXiv preprint arXiv:2006.10842*, 2020.

[96] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[97] Zhou Yang, Long Hoang Nguyen, Joshua Stuve, Guofeng Cao, and Fang Jin. Harvey flooding rescue in social media. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2177–2185. IEEE, 2017.

[98] Neil Yeung, Jonathan Lai, and Jiebo Luo. Face off: Polarized public opinions on personal face mask usage during the covid-19 pandemic. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 4802–4810. IEEE, 2020.

[99] Jie Yin, Sarvnaz Karimi, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. Using social media to enhance emergency situation awareness. In *Twenty-fourth international joint conference on artificial intelligence*, 2015.

[100] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8 (4):e1253, 2018.

[101] Caleb Ziems, Bing He, Sandeep Soni, and Srijan Kumar. Racism is a virus: Anti-asian hate and counterhate in social media during the covid-19 crisis. *arXiv preprint arXiv:2005.12423*, 2020.

[102] Lei Zou, Nina SN Lam, Shayan Shams, Heng Cai, Michelle A Meyer, Seungwon Yang, Kisung Lee, Seung-Jong Park, and Margaret A Reams. Social and geographical disparities in twitter use during hurricane harvey. *International Journal of Digital Earth*, 12(11):1300–1318, 2019.

VITA

Md Yasin Kabir was born in a small beautiful country named Bangladesh. He received his bachelor's degree in Computer Science and Telecommunication Engineering from Noakhali Science and Technology University in 2014. Shortly after completing his bachelor's degree, Yasin joined as a lecturer in the department of CSTE of Noakhali Science and Technology.

Yasin obtained his Ph.D. in Computer Science in July 2022 from Missouri University of Science and Technology under the supervision of Dr. Sanjay Madria. Since 2017, he worked on data analytics and machine learning. His primary research interest was analyzing and extracting information from social media for disaster management. He was very passionate to solve problems in other domains which leads him to various Kaggle competitions where he became a competition master in 2020. During his Ph.D. he worked as an intern in Verisign and Nvidia to solve different problems leveraging machine learning and GPU-based data processing.

Yasin was very passionate about gardening. In 2021 he grew 11 different types of vegetables in his garden. He had a great deal of interest in photography and blogging.