IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 July 2022, accepted 15 July 2022, date of publication 19 July 2022, date of current version 26 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192417

RESEARCH ARTICLE

Sentiment-Based Spatiotemporal Prediction Framework for Pandemic Outbreaks Awareness Using Social Networks Data Classification

NOHA GAMAL[1], SAMY GHONIEMY[2], (Member, IEEE), HOSSAM M. FAHEEM[3], AND NOHA A. SEADA[3]

[1]Faculty of Information Technology and Computer Sciences, Nile University, Giza 12677, Egypt
[2]Faculty of Informatics and Computer Science, British University in Egypt, El-Sherouk City, Cairo 11837, Egypt
[3]Faculty of Computer and Info. Sciences, Ain Shams University, Giza 11566, Egypt

Corresponding author: Noha Gamal (ngamal@nu.edu.eg)

ABSTRACT According to the World Health Organization, several factors have affected the accurate reporting of SARS-CoV-2 outbreak status, such as limited data collection resources, cultural and educational diversity, and inconsistent outbreak reporting from different sectors. Driven by this challenging situation, this study investigates the potential expediency of using social network data to develop reliable early information surveillance and warning system for pandemic outbreaks. As such, an enhanced framework of three inherently interlinked subsystems is proposed. The first subsystem includes data collection and integration mechanisms, data preprocessing, and hybrid sentiment analysis tools to identify tweet sentiment taxonomies and quantitatively estimate public awareness. The second subsystem comprises the feature extraction unit that identifies, selects, embeds, and balances feature vectors and the classifier fitting and training unit. This subsystem is designed to capture the most effective linguistic feature combinations with more spatial evidence by using a variety of approaches, including linear classifiers, MLPs, RNNs, and CNNs, as well as pre-trained word embedding algorithms. The last is the modeling and situational awareness evaluation subsystem, which measures temporal associations between pandemic-relevant social network activities and officially announced infection counts in the most hazardous geolocations. The proposed framework was developed and tested using a combination of static datasets and real-time scraped Twitter data. The results of these experiments showed the remarkable performance of the framework in assessing the temporal associations between public awareness and outbreak status. It also showed that the Decision Tree Classifier with Unigram+TF-IDF feature vectors outperformed other conventional models for sentiment classification and geolocation classification with an accuracy of 94.3% and 80.8, respectively. As indicated, conventional machine learning algorithms didn't achieve a precision of more than 80%, while, for instance, MLP with self-embedding layer, Word2Vec, and GloVe pre-trained word embedding resulted in very poor accuracy of 10%, 36%, and 32%, respectively. However, adding the PoS tag one-hot encoding embedding increased the validation accuracy from 36% to approximately 89%, while the best performance for the second subsystem was achieved by Bi-LSTM with RoBERTa word embedding, with an accuracy of 96%. The achieved results reveal that the proposed framework can proactively capture the potential hazards associated with the prevalence of infectious diseases as an effective early detection and info-surveillance awareness system.

INDEX TERMS Social network analysis, sentiment analysis, pandemic outbreak, geolocation prediction, data correlation, spatiotemporal analysis, outbreak awareness, pandemic data classification.

I. INTRODUCTION

According to the World Health Organization, Coronavirus disease (COVID-19) is an infection caused by the SARS-CoV-2 virus. In the first phases of the pandemic, numerous research studies focused on studying the coronavirus social network to better understand the outbreak's effects on society and mitigate the spread of the outbreak's infodemics [1]–[9]. Many of these studies explore the importance of employing social network analysis-based models and frameworks in a variety of applications. Social networks such as Twitter and Facebook have evolved into a

The associate editor coordinating the review of this manuscript and approving it for publication was Ines Domingues.

valuable method for analyzing user sentiment for a variety of purposes [9]–[16]. In addition, locating social media users is essential for a variety of reasons, including providing area-specific services and recommendations, detecting earthquakes, managing natural disasters, investigating crimes, assessing demographic data, and healthcare management [9]–[17]. This is especially important when dealing with disease epidemics such as COVID-19. So, surveillance and early recognition are crucial for mitigating infectious disease outbreaks. Recent research has revealed that the spread of an infectious disease is directly associated with population geolocations and mobility [1]–[8]. Meanwhile, it was reported that using social networks for actionable disease surveillance and outbreak control has proven to have a high potential for success, as identified in exploratory studies [9]–[11]. The problem is that social network data is dynamic, vast, and unstructured, requiring sophisticated algorithms and computational linguistics [12]. However, compared to traditional assessments and clinical reports, which require a lot of time and finance to gather data, social media data can be scraped from many social network platforms immediately and at a lower cost [13]. Yet, social media will never replace conventional surveillance [16], but it may provide supplementary data when inadequate or can be used in conjunction with traditional data [17]. The spatial and temporal data may also be used to investigate the spatiotemporal dynamics of disease transmission. As epidemics such as COVID-19 become widely reported on social network platforms, these sources experience and express a variety of perspectives, opinions, and emotions during various outbreak-related events. Analysis of these sentiments, along with some demographic features, will thus provide interesting findings regarding understanding the outbreak spread track. Additionally, several theories, speculations, and disinformation circulated on social media. In addition to the tweet text and metadata, Twitter allows users to broadcast their geo-location through the Global Positioning System function. Only a small percentage of users decide to reveal their location, while all other Twitter users decide to conceal their geolocation information to protect their privacy or to avoid harassment, trailing, or trolling [7]. Identifying the geolocation of social media users is critical instead of providing area-specific functions and suggestions, especially for healthcare management [18]–[19], and particularly during an epidemic outbreak [20]. Accordingly, this research investigates how social network data may be used to develop reliable, early information surveillance and warning system for pandemic outbreaks. This paper proposes deep social network data analysis concerning time to enhance pandemic outbreak awareness from two perspectives: a) chronological sentiment analysis and b) chronological spatial analysis and prediction. As part of this study, we use linguistic characteristics to capture sentimental and spatial variances from tweets. This research conducts a social network data analysis that considers more than 577 thousand geotagged, COVID-19-related tweets to evaluate the viability. The framework proposed in this study addresses three aspects using its proposed three subsystems. The first subsystem includes data collection and integration mechanisms, data preprocessing, and hybrid sentiment analysis tools. The second subsystem comprises the feature extraction unit that identifies, selects, embeds, and balances feature vectors and the classifier fitting and training unit. The last subsystem is the situational awareness evaluation subsystem that measures temporal associations between pandemic-relevant social network activities and officially announced infection counts in the most hazardous geolocations. Hence the first aspect is proposed as a quantitative analysis methodology for estimating public awareness of COVID-19 using the identification of sentiment taxonomies of tweets through hybrid sentiment analysis tools. The second aspect proposes a methodology for geographical location prediction using only Twitter textual data and predicted sentiments. Once the scraped tweets' locations are reliably predicted, the third aspect is incorporated into the proposed system. This subsystem is also implemented as a crowd situational awareness evaluation methodology that is devised to quantify the temporal association between the formation of a certain public sentiment and the corresponding COVID-19 active case count in the most hazardous geolocations by computing the casual synchrony between the tweet activities associated with sentiment scores of COVID-19 related tweets and the infected cases count. The entire framework architecture, methodologies, procedures, and algorithms are explained in more detail in the methodologies section. Because the suggested system is designed to rely on simple text inputs, it may be more easily adapted to address new areas and languages, like Arabic-tongued areas, by imposing the appropriate algorithms. Consequently, we employ various strategies for improving the text representation components and the overall model design to overcome many Natural Language Processing (NLP) related problems, including textual classification and sentiment analysis, using deep learning techniques. To improve the location prediction accuracy, we evaluated alternative models, including linear classifiers, Convolutional Neural Networks (CNN), Multi-Layer Perceptron (MLP), and Recurrent Neural Networks (RNN), empowered with word embedding algorithms such as self-embedding, word2vec, GloVe, and FastText, in addition to transformer-based word embedding such as BERT and RoBERTa. From many perspectives, this study examines the significance of employing social network data and machine learning/deep learning classification algorithms to improve the awareness prospect of infectious disease as a proactive info surveillance framework. The detailed abbreviations and definitions used in the paper are listed in Table 1. The remaining of this paper is organized such that related studies and a literature survey are presented in section II. A detailed description of the proposed framework, implementation procedures, and methods are described in section III. While section IV included our main findings, results, and

TABLE 1. List of abbreviations and acronyms used in the paper.

Abbreviation	Definition
SARS-CoV-2	severe acute respiratory syndrome coronavirus 2
NLP	Natural Language Processing
CNN	Convolutional Neural Networks
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Networks
GPS	Global Positioning System
K-NN or KNN	K-Nearest Neighbors
CRF	Conditional Random Fields
HMM	Hidden Markov Model
NB	Naïve Bayes
SVM	Support Vector Machines
DNN	Deep Neural Networks
NTLK	Natural Language Processing Tool Kit
NBM or MNB	Naïve-Bayes Multinomial
ILI	Influenza Like Infectious
LDA	Latent Dirichlet Allocation
VADER	Valence Aware Dictionary and sEntiment Reasoner
LSTM	Long-Short Term Memory
Bi-LSTM	Bi-directional Long-Short Term Memory
GRU	Gated Recurrent Unit
BERT	Bidirectional Encoder Representations from Transformers
GNN	Graph Neural Networks
SGC	Simple Graph Convolution
DT	Decision Tree
HLPNN	Hierarchical Location Prediction Neural Network
NER	Named Entity Recognition
RoBERTa	Robustly Optimized BERT
TLCC	Time-Lagged Cross-correlation
WTLCC	Windowed-Time-Lagged Cross-correlation
BoW	Bag of Words
TF-IDF	Term Frequency – Inverse Document Frequency
GBM	Gradient Boosting Machine
ANN	Artificial Neural Network
MLM	Masked Language Modelling
NSB	Next Sentence Prediction
BPE	Byte-Pair Encoding

discussions, limitations, strengths, and complexities are presented in section V. Moreover, conclusions and future studies are presented in Section VI.

II. RELATED STUDIES

When COVID-19 initially emerged, the most pressing concern was how to mitigate the infection and protect billions of citizens globally while not compromising the international economy, which could be badly disrupted if governments imposed absolute lockdown and quarantine policies. Many countries worldwide endured and continue to suffer economic disruptions due to national/regional lockdown periods ranging from a few days to hundreds of days [21]–[22]. For the economy to recover, people should be allowed to move freely. This necessitates that authorities be able to swiftly track the potential contacts of any discovered infected cases [16]. In response to the development of coronavirus disease, numerous governments, healthcare national organizations, and institutions have initiated a contact tracing network analysis over billions of Global Positioning System ''GPS'' human mobility data points to monitor the evolution of the disease contact network. To effectively combat epidemics, governments may be justified in restricting some fundamental rights and liberties. However, such restrictions must adhere to applicable fundamental rights standards and privacy regulations [23]. Numerous individuals and social organizations voiced privacy rights concerns in conjunction with the use of contact tracking programs, although a portion of the population accepted a limited breach of privacy for the sake of health protection. Since the infectiousness period begins before the start of symptoms, despite the reluctance with which such apps were accepted, they were of little assistance in preventing the spread of disease [16], [24]. In other words, the latency period is shorter than the incubation period, and infection occurs prior to the onset of symptoms. Moreover, on average, the patient is at its most contagious stage, i.e., the peak of infectiousness, two days before the onset of symptoms, according to a study conducted at the outbreak's onset [25]. These facts are essential for understanding the widespread of COVID-19. They imply that by the time a patient develops symptoms, he or she has now transmitted the infection to the greater part of infected individuals during the preceding two days. Even if the patient isolates themselves after developing symptoms, most infection transmissions have already emerged [24]. So, diverse studies have employed social network-based geo-positioning for tracing social contacts in order to examine the likely progression of the infection as a safer and more widely accepted alternative to sensor-based geo-positioning tracking applications. Social media data broadcasting via social networks can help with situational awareness, information dissemination, and monitoring of different activities [10]–[16] [21]. Recent research investigates the importance of using social network analysis-based methodologies and frameworks for numerous applications. Sentiment analysis on social media platforms like Twitter and Facebook has emerged as a powerful and effective approach to studying user sentiments in a variety of contexts [9]–[16].

Furthermore, identifying the location of social media users is crucial to providing area-specific facilities and recommendations, earthquake surveillance, natural disaster governance, crimes, demography analysis, and health care management [9]–[17], especially when assessing, interpreting, and reacting to an epidemic outbreak, such as COVID-19. Therefore, has integrated NLP and machine learning with social network data to improve textual data analysis. Multiple approaches use textual content, information diffusion, or emotion patterns to detect emerging events [25]–[29]. Such studies considered COVID-19's social impact and explored how social media may help the government evolve control policy [30]. Also, in the field of tracking and monitoring any human-threatening disaster, a typical application tracks sentiment fluctuations depending on the population's geolocation, attempting to improve natural disaster early warning systems [31]–[32]. A survey of related studies in the fields of sentiment analysis and location prediction is presented in the following sub-sections.

A. SOCIAL MEDIA FOR SENTIMENT ANALYSIS

Sentiment analysis has been extensively studied in the scope of online content reviews to obtain concise user views on

a variety of attributes. Recognizing such sentiments from social networking websites such as Facebook and Twitter can aid emergency responders in comprehending the network's dynamics, such as the primary users' concerns, panics, and the emotional impact of member interactions. Generally, three layers of feature extraction are used in sentiment classification: word, phrase, and document [33]. At the moment, sentiment analysis can be performed using three methods: lexicon-based, machine-learning-based, or hybrid [34]. The earliest approaches to sentiment analysis relied on lexicons. Their approach categorizes them as either dictionary-based or corpus-based [35]–[36]. The former version uses a word dictionary, such as SentiWordNet or WordNet. However, sentiment analysis tasks based on corpus do not use a pre-defined lexicon but rather statistically analyze a collection of documents using machine learning techniques such as K-Nearest Neighbors (K-NN) [37], Conditional Random Fields (CRF) [38], and Hidden Markov Model (HMM) [39], which direct this field to machine learning-based sentiment analysis models. According to machine learning [40], there are two types of sentiment analysis: conventional and deep learning approaches. The Naïve Bayes (NB) classifier [41], the Maximum Entropy classifier [42]–[43], and Support Vector Machines (SVM) are all examples of conventional models. Sentiment analysis can also be performed using deep learning models such as Convolutional Neural Networks (CNN), Deep Neural Networks (DNN), and Recurrent Neural Networks (RNN), where deep learning models have the potential to outperform conventional models [44]. According to a review of studies based on traditional statistical approaches on social networks and COVID-19 presented by [35], public sentiments were the most prevalent theme among many identified themes, accounting for the majority of articles. As reported in [35], Sina Weibo was the most popular social media network following Twitter. The use of machine learning by epidemiologists to sift through huge amounts of digital data is described in [45]. They explored how to employ natural language processing and machine learning to analyze massive datasets for population-level mental health research. The results obtained by [46] indicate that the majority of people viewed the epidemic positively and supported the government's or local authorities' actions. While the counts of infected and deceased individuals continued to grow, the population's mental power remained unaffected. They reached their conclusions by developing a machine learning-based sentiment analysis (NTLK, TEXT BLOB, and Naïve-Bayes Classifier) considering people's responses to the government or local authority decisions made during the outbreak. Their proposal recommended a way of automatically categorizing tweets as positive, negative, or neutral. Using the same technologies, researchers in [47] analyzed tweet sentiments concerning both polarity and subjectivity. They concluded that, for polarity, over 36% of people tweeted positive messages, while only 14% tweeted negative messages. Nevertheless, neutral polarity for both ''coronavirus'' and ''covid-19'' terms was high, at around 50%. Additionally, concerning subjectivity, overall, around 64% of the entries for both keywords (''COVID-19, coronavirus''), were objective. Meanwhile, 22% were subjective in expressing their views and opinions. Finally, 14% of tweets are neither subjective nor objective.

Regarding the classification of tweets into two classes, such as in [48], the authors discovered that the k-NN classifier outperformed many other machine learning classifiers, including Naïve-Bayes Multinomial (NBM) Modal, NB, and SVM. Furthermore, the authors in [49] used Twitter data to forecast the weekly status of the influenza-like-infectious (ILI) infected population in the United States using a multi-layer perceptron with a backpropagation technique for independent test sets. The deep granulator methodology was also used to improve precision using supervised machine learning methods for recognizing personal health experiences [50].

A similar study [51] examines the possibility of leveraging Twitter data to alert the public about the US COVID-19 pandemic. By ''crowdsourcing'' public opinions, this strategy allows the public and private sectors to timely uncover hazards and prepare for pandemics. Random Forest, Logistic Regression, Support Vector Machines (SVM), and Naïve Bayes (NB) are used to classify tweets. Unlike supervised algorithms, unsupervised algorithms do not require labeled data sets [52]. Unsupervised classification algorithms seem more tempting, yet they may be more challenging to obtain equal accuracy. In [53], the authors used a set of tools (Tweepy Python module, Twitter's search API, and PostgreSQL database) and specified search phrases (''corona,'' ''2019-nCov,'' and ''COVID-19''). They studied public English-language tweets in the period from February 2nd, 2020, to March 15th, 2020, for word frequencies of unigrams and bigrams. They used Latent Dirichlet Allocation (LDA) to model the topics discussed in tweets. They also conducted sentiment analysis and estimated the count of likes, retweets, and follows for each subject. In another study [54], the authors analyzed COVID-19-related tweets submitted by individuals to understand what common subjects and topics emerged, as well as how sentiments of individuals evolved over time from before to after the announcement of COVID-19. To find patterns and use Valence Aware Dictionary and sEntiment Reasoner (VADER) to get sentiment scores and to look at weekly trends over a period of 17 weeks were utilized. Initial Twitter reactions were negative when it came to the influence of COVID-19 on the healthcare sector, including hospitals, clinics, and front-line employees. However, over the weeks, these reactions changed to positive sentiment. Similarly, [55] focused on the information flow on Twitter during the Corona Virus outbreak. For sentiment and topic modeling, LDA is used in post-processing to analyze coronavirus tweets. The study revealed that information flow was accurate and credible during the coronavirus outbreak. It revealed the prevalence of negative sentiments like fear and positive ones like trust.

Many researchers have tackled employing Deep Neural Networks and Recurrent Neural Networks Algorithms with text representation, tokenization, and word embedding. For example, in the methodology presented by [56], the authors obtained an accuracy of 84.5% when using Long-Short Term Memory (LSTM) as a supervised Deep Learning Algorithm to evaluate the sentiment of a tweet. Such that, textual input features such as tweet text, country name, and date were used to predict the tweet text polarity. Initial polarity was calculated using the VADER module. The authors in [57] also proposed a method to examine citizens' reactions to the new coronavirus and their views on subsequent national actions. LSTM, Bi-LSTM, and GRU are used with different word embedding like GloVe and BERT to estimate sentiment polarity and emotions from retrieved tweets. The highest accuracy was obtained when using LSTM+Glove: Training accuracy was 93%, validation accuracy was 83%, and testing accuracy varied from 69% to 84% when applied to different datasets. Based on the outcomes of this survey, we can conclude that using deep learning models instead of traditional machine learning algorithms to solve sentiment classification problems does not result in significant precision improvements. It also consumes a lot of computing power. As a result, our proposed approach relies solely on traditional machine learning algorithms to compute sentiment scores for tweets.

B. SOCIAL MEDIA FOR SPATIAL ANALYSIS AND PREDICTION

Identifying the user's current location and the location of a tweet are two distinct concerns that can be tackled using geolocation prediction on Twitter data. The problem of determining a user's geolocation can be divided into two sub-problems: a user's home location and a user's current location. Additionally, the tweet geolocation prediction problem may be broken down into two sub-problems: predicting where a tweet came from (i.e., where it was sent from) and extracting the locations mentioned in a tweet. Location-based social networks are likewise becoming increasingly popular [58]–[60]. Earlier work on tweet geolocation prediction employed machine learning [61]. These methods resulted in poor performance because of the massive amount of data available on Twitter that needed to be analyzed and trained. Current research has changed the attention from ML to DL for Twitter user location forecast. Various approaches and strategies have been employed during the last few years to improve the accuracy of location prediction methodologies and algorithms. Various research studies have used metadata, such as temporal information, to conclude the location of users, as presented by the research in [62]. Some have just used the tweets' content as in [63]–[65]. Others have studied users' social networks [66]. User account data has also been effective [67]–[68], while others have used a hybrid approach [69]. However, most research studies continue to use the text of the post or tweet as a major input for their study. Over the years, diverse features and indicator types have been employed to predict Twitter users or tweet locations. These may include message text or context, the user's followers and friends network, the user's profile metadata, and the user's and tweets' geotags, among many other indicators. As a result, natural language processing (NLP) via machine learning, deep learning, and probabilistic approaches to locating databases [67] has been widely used. To match the actual location to keywords, [60] employs cosine similarity. While performing content analysis, evaluating all possible occurrences of the geographical entities expressed in the message is important. The user's location or even the tweet location cannot be automatically deduced from messages [70]. Employing feature extraction using N-Gram and Bidirectional Encoder Representations from Transformers (BERT) Embedding, the authors in [71] have tested ensemble models based on meta-learners and deep neural networks, such as XG-Boost, LSTM, and Character-Level-CNN, respectively, with the highest, obtained an accuracy of 53.88% for the localization of Swiss German Tweets. Certain methods have been suggested to exploit URL linkages within the text to infer the users' location. A free online query tool that correlates geographical location with IP addresses and domain names is used in [72] to predict the country-level location, featuring location-indicative words, hashtags, user mentions, and metadata in the tweet text. In another study, the authors used Naive Bayes and Logistic Regression to locate tweets using geo-location-specific phrases and hashtags. Then, a year later, they presented a stacking-based technique [73]–[74]. That combined tweet content and metadata. They also considered the influence of using tweets with no location tagging, users' language, and user-declared metadata on the geolocation forecasting task and how user behavior varies by region. A user's social network, which consists of his followers or friends, has been demonstrated to be a reliable indicator of their home address. Some research [75]–[76] has demonstrated that if two users live in the same city, they are more likely to contact each other regularly and vice versa. Using a graphical representation of tweeting users' networks and with the help of Graph Neural Networks (GNN), specifically the ready-implemented Simple Graph Convolution (SGC), and MLP (Multi-Layer Perceptron), the authors in [77], were able to predict the geolocation of a tweeting user. Additionally, they explored many differences between SGC and MLP in order to analyze the results. SGC, for example, is a ''GNN-based'' model that includes tweets' text and node location as features, whereas MLP embeds tweet text and locations independently. For SGC, the authors of [77] achieved a maximum accuracy of 62.5 percent. Their results also implied that such models are prone to adversarial attacks, which is challenging for all GNN-based models. According to [78], the more influential a user is, the more diverse their followers and friends are from all over the world, which might negatively affect the prediction accuracy. The authors have obtained a similar result in [66] as they made it evident that an ordinary user's network is optimum for inferring position up to the third depth. The authors in [79] proposed an approach to forecasting the city-level geolocation of tweets

collected over a period of 30 days using a combined model of CNN and Bi-LSTM. This is accomplished by mining features contained within the tweets and the tweet metadata, for instance, the tweet text, user screen name, and user profile location. By merging two neural networks, CNN and Bi-LSTM, and leveraging Deep Learning methods with word embedding, the authors suggested a method to handle the task of Tweet geolocation prediction. Word embedding was employed using word2vec (a Google-trained algorithm). The proposed method had a classification accuracy of 92.6 percent. Most smartphones now come with GPS built-in, allowing geo-satellites to determine the user's exact location using latitude and longitude coordinates. Due to privacy concerns, users frequently disable this function, making geotagging a challenging feature. In the discipline of geography and land surveying, Vincenty's geometric median is used to estimate a Twitter user's location based on their last certain count of geotagged tweets within an area of 15 km radius as presented by [76] and [80]. A similar study was proposed by the authors in [81], presenting a method for calculating the distance between tweets from the same user by predicting the user's home location using tweet-relevant information. The first tweet of each user is considered the individual's baseline home location. Classic machine learning algorithms such as Support Vector Machine (SVM), Linear Regression, Linear Model Theil Sen Regressor, Decision Trees, Linear Model SGD Regressor, Neural Networks, K-nearest neighbor (KNN), RandomForest, Gradient Boosting Regressor were applied. The highest accuracy was obtained using DT, at 85%. Using Deep Learning with word embedding, sentiment analysis, and featuring tweet content and tweet geolocation (longitude and latitude), the authors in [82] proposed DeepGeoloc with a max accuracy of 58.4%. Long-Short Term Memory (LSTM) was tested with different word embedding methods, such as Word2vec, FastText, and Char2vec. The authors of [83] have combined tweets' content and user-profile metadata into one model using CNN. Their suggested approach was more accurate, although the results showed some bias that affected the resultant prediction accuracy. Furthermore, their study [84] proposed a hierarchical location prediction neural network (HLPNN).

C. SOCIAL MEDIA FOR TEMPORAL PUBLIC AWARENESS

Not so many studies have been conducted on using Twitter data to computationally quantify the public awareness measures regarding the recent COVID-19 outbreak. Some studies have employed manual surveying tools to collect observations that might correlate public awareness with the outbreak variations during a certain period. The authors in [51] have employed NLP and conventional machine learning models to classify tweets contents into ''signal'' or ''not signal'', while the first-class indicates that the Twitter user recognizes the COVID-19 outbreak risk in USA. The ''signal'' tweets volumes were compared to the counts of COVID-19 active cases. The authors indicated that the emergence of a ''signal'' classified tweets volume had a leading time of 16 days. Another study [109] collected tweets about the 2020 COVID-19 pandemic in the United States expressing the most typical symptoms of COVID-19, such as cough and fever. Through simple exploratory data analysis and visualizations, the authors have indicated a 5 to 19-day lag between increases in the number of symptom-reporting posts on Twitter and officially reported confirmed cases by analyzing the fluctuation in Twitter activities at the state level. Similarly, the authors in [110] have employed the Named Entity Recognition (NER) method to measure the lead time of the COVID-19 public awareness. This study has stated a time lead of 6-27 days. Even though these studies used not considerably suitable computational methods, these presented social data analysis which reflected the importance of such type of data in the field of pandemic outbreak awareness. The most relevant literature in comparison to the proposed framework is summarized in Table 2 below. This summary describes the literature dataset, features, methods, evaluation metrics, added values, and limitations. As disclosed in the previous survey, using social media data to track trends and gauge public opinion about pandemic-response strategies can be a cost-effective, timely, and informative strategy. So, we found that developing a general, reliable, proactive awareness system for pandemic outbreaks that employs more suitable methods is of great importance, mainly through predicting the outbreak's most affected geolocations. Accordingly, the predicted and known user and tweet locations will help in preventing an epidemic of hazardous diseases, as the disease can be contained in its earlier emergence, enabling health and governmental organizations to plan the best infection control measures and policies. As such, we were motivated to propose our enhanced framework of the three inherently interlinked subsystems for identifying sentiment taxonomies, inferring the most hazardous geolocation, and evaluating the temporal associations between public awareness and outbreak status as described earlier, such that, based on our review of the literature, very few related research evidently presented a methodology to measure the potential expediency of using social network data to develop reliable early information surveillance and warning system for pandemic outbreaks, especially in the way we designed and developed our framework.

III. METHODS AND PROPOSED METHODOLOGY

A. RESEARCH GOALS AND QUESTIONS

When conventional surveys and clinical report data are insufficient, social media data may source additional data easily and effectively. Such that, our research suggested a methodology to evaluate how social network data such as tweets, the user or tweet geolocations, and public sentiments, among other features, may be utilized to construct an accurate pandemic information surveillance system. Disease transmission's spatial and temporal dynamics may be explored using social media data. From many perspectives, this study examines the significance of employing social network data and machine learning/deep learning classification algorithms to

TABLE 2. Literature survey summary.

Objective	Research Work	Dataset	Features	Approach	Evaluation Metrics	Value-Added	Limitation
Sentiment Analysis	Rijwan Khan, 2020	**Source:** Twitter **Samples Count:** 50K **Geo-Area:** India	**Input:** Tweets Content **Output:** Polarity and Subjectivity of public opinions	Naïve-Base Classifier, NLTK	Qualitative Visualizations and Plots.	Using NLTK reduces the complexity of the implementation	Evaluating the proposed solution using Qualitative visualizations and interpretations weakens the results' reliability, especially when not compared to related literature.
	Muvazima Mansoor, 2020	**Source:** Twitter **Samples Count:** ~280K **Geo-Area:** USA, India, Brazil	**Input:** Tweets Content and Metadata **Output:** Tweet Sentiment	MLP, LSTM	Accuracy: MLP: 76% LSTM: 84.5%	Employing DL automates much of the feature extraction process, eliminating some of the manual human intervention required. It also enables the use of large data sets, which increases the proposed system scalability	Using DL methods for sentiment analysis added more complexity and increased resource overhead without remarkable improvement in accuracy.
	Mostafa Al-Emran, 2020	**Source:** Twitter **Samples Count:** ~7M **Geo-Area:** Worldwide	**Input:** Tweets Content **Output:** n-gram-based Exploratory Data Analysis and Insights	Evolutionary K-means over time Clustering	Qualitative Visualizations and Plots.	Using a vast amount of input data increased the reliability of the obtained results	The study doesn't propose any evaluation metrics for the obtained results. Results were not compared to any similar literature.
	Ali Shariq Imran, 2020	**Source:** Twitter **Samples Count:** ~27K (Scrapped) (~460K) Kaggle Dataset), (Sentiment140) **Geo-Area:** Worldwide	**Input:** Tweets Content and Metadata **Output:** n-gram-based Exploratory Data Analysis and Insights	Deep Learning (LSTM, BiLSTM, GRU) with word embedding (FastText, Glove, Glove-Twitter, BERT), sentiment analysis	Accuracy, F1 Score: range from 60% to 82% for different model stages and different methods combinations	This study compares the performance metrics of the system using different datasets and data variations, which shows the expandability of the proposed system	This work compares the performance of different embedding and uses different variants of RNN; however, it does not assess the performance of other deep neural networks like convolutional neural networks (CNN) and their variants.
	Kamaran Manguri, 2020	**Source:** Twitter **Samples Count:** ~530K **Geo-Area:** Worldwide	**Input:** Tweets Content **Output:** Sentiment-based Exploratory Data Analysis and Insights	TEXTBLOB, NLTK	Qualitative Visualizations and Plots	Using NLTK reduces the complexity of the implementation	Evaluating the proposed solution using Qualitative visualizations and interpretations weakens the results' reliability, especially when not compared to related literature.
	Zhe Zhang, 2020	**Source:** Twitter **Samples Count:** ~80M **Geo-Area:** USA	**Input:** Tweets Content and Tweets Geolocation **Output:** Sentiment-based Exploratory Data Analysis and Insights	Lexicon-based Sentiment Analysis	Qualitative Visualizations and Plots	In addition to employing the -based sentiment analysis and topic classification models to mark stress signs within a certain population, the proposed study adopts fuzzy assessment assessment method which increased the reliability.	Evaluating the proposed solution using Qualitative visualizations and interpretations weakens the results' reliability, especially when not compared to related literature.
	Ranganathan Chandraskran, 2020	**Source:** Twitter **Samples Count:** ~13.9M **Geo-Area:** Worldwide	**Input:** Tweets Content **Output:** Sentiment-based Exploratory Data Analysis and Insights	LDA, Lexicon-based Sentiment Analysis and Topic Classification	Qualitative Visualizations and Plots	The research presented an extensive social data analysis which reflected the importance of such type of data in the field of pandemic outbreak awareness	The research did not consider any geographical boundaries when examining the tweets. This makes it unable to capture different topics/sentiment of the population in a specific geolocation. So the results are too general to be describing a limited geo-area.
	Bishwo Prakash Pokharel, 2020	**Source:** Twitter **Samples Count:** ~615 **Geo-Area:** Nepal	**Input:** Tweets Content **Output:** Polarity and Subjectivity of public opinions	TEXTBLOB, NLTK	Qualitative Visualizations and Plots	Using NLTK reduces the complexity of the implementation	Evaluating the proposed solution using Qualitative visualizations and interpretations in addition to the limited tweets counts weakens the results' reliability, especially when not compared to related literature.
	CHIUNG CHING HO, 2020	**Source:** Twitter **Samples Count:** ~370K **Geo-Area:** USA	**Input:** Tweets Content and Tweets Geolocation **Output:** Sentiment-based Exploratory Data Analysis and Insights	CNN, BiLSTM, VADER	Accuracy, CNN(binary): 85.1% CNN(Multi): 84.2% BiLSTM(binary): 86.2% BiLSTM(Multi): 87.7%	The authors have employed SHapley Additive exPlanations (SHAP) to weight the effect of selected features on the classification model. Such that the higher the mean SHAP values of a feature, the higher the contribution of that feature to the model.	Using DL methods for sentiment analysis added more complexity and increased resource overhead without remarkable improvement in the accuracy.
Location Prediction	W.Gad, 2021	**Source:** Twitter **Samples Count:** ~380K **Geo-Area:** USA	**Input:** Tweets Content, Tweets metadata, Sentiment, Tweet location **Output:** User Location	LR, SGD, Theil Sen, SVM, DT, RF, NN, KNN, GBR	Accuracy@161: 85% for the Decision Tree model	Regression methods employed in this study are most likely to be more indicative when it comes to predicting catastrophic geolocation.	This study has employed an outdated instead of a recent dataset and compared the resulted accuracy with related work published in the period from 2010 to 2015.
	Zahid A Butt, 2020	**Source:** GeoText and Twitter-US **Samples Count:** ~10K, 450K **Geo-Area:** USA	**Input:** User Followers and Friends Network + tweet content embedding **Output:** User Location	GNN, SGC, MLP	Accuracy: SGC, 62.5%	Authors have examined how different NN models deal with text embedding	Presented approaches lack model transparency and relevant explanations of model behavior and prediction findings, preventing them from being used in safety-critical applications.
	Sarra Hasni, 2020	**Source:** Twitter **Samples Count:** ~5M **Geo-Area:** USA, UK	**Input:** Tweet content, Tweet Geolocation **Output:** User Location	Long Short Term Memory (LSTM) with word embedding, Word2vec/FastText /char2vec	Accuracy: 58.31%	The presented framework scalability option is very interesting	This work compares the performance of different embedding and uses different variants of RNN, however it does not assess the performance of other deep neural networks like convolutional neural networks (CNN) and their variants.
	Rhea Mahajan, 2021	**Source:** Twitter **Samples Count:** ~45K **Geo-Area:** India	**Input:** Tweet content, user name, User Geolocation **Output:** Tweet Location	CNN and BiLSTM. With Word Embedding, word2vec, GLOVE	Accuracy: 92.6%	The presented algorithm is considerably efficient from the computational space point of view, because of the reduced size and dimensionality of the feature vector employed by CNN and BiLSTM.	As proposed by the authors, the combination of CNN and RNN layers in one algorithm caused very high computational time complexity, which may reduce its efficiency and reliability.
Public Awareness Analysis	Erfaneh Gharavi, 2020	**Source:** Twitter **Samples Count:** ~40K **Geo-Area:** USA	**Input:** Tweet Volume **Output:** lag/lead time between Tweets counts time series and no. of infected cases time series	Exploratory Data Analysis using data visualization and interpretation	Time Period: a temporal lag between the rises in the number of symptoms reporting tweets and officially reported positive cases, which varies between 5 to 19 days.	Although this study used primitive computational methods, it presented social data analysis which reflected the importance of such type of data in the field of pandemic outbreak awareness.	The achieved results were based on very primitive computational methods, which may differ if complex methods such as classifiers or correlators were employed.

TABLE 2. ***(Continued.)*** **Literature survey summary.**

	Lei Gao, 2021	**Source:** Twitter **Samples Count:** ~4M **Geo-Area:** USA	**Input:** Tweet Volume, Tweet content **Output:** Tweet Class, lag/lead time between Tweets counts time series, and no. of infected cases time series	SVM, RF, LR, NB + TF-IDF, cross-correlation analysis	Accuracy: LR, 93% Time Period: a temporal lag between the rises in the number of symptom reporting tweets and officially reported positive cases of 16 days.	Although this study used primitive computational methods, it presented social data analysis which reflected the importance of such type of data in the field of pandemic outbreak awareness.	Manual labeling of the very limited 5K tweets out of 4M volume can be considered not appropriate as a training set. Additionally, many assumed parameters, such as the critical signal value, need to be discussed and explained.
	I Kit Cheng, 2021	**Source:** Twitter **Samples Count:** ~ 771K **Geo-Area:** UK	**Input:** Tweet Volume, Tweet content **Output:** Tweets topic modeling, user location, lag/lead time between Tweets counts time series, and no. of infected cases time series	Named Entity Recognition, Spearman's correlation Analysis	Accuracy: NER for topic modeling and location prediction: 81% Time Period: a temporal lag between the rises in the number of COVID reporting tweets and officially reported positive cases, which varies between 6 to 27 days.	Even if this study used not considerably suitable computational methods, it presented social data analysis which reflected the importance of such type of data in the field of pandemic outbreak awareness.	The employed methodology and achieved results for the location prediction module need to be reviewed and improved for more generalization and scalability.
Proposed Framework	Sub-system 1: Sentiment Analysis and Feature Selection	**Source:** Twitter **Samples Count:** ~ 577K **Geo-Area:** USA, UK, India, Australia, Canada	**Input:** Tweet content, Tweet Metadata **Output:** Tweets sentiment, data features required for next sub-systems	Hybrid Sentiment Analyzer (VADER)+(MNB /DT/ET/RF/Ada/GBM)	Accuracy: DT, 94.3%	Discussed in detail in the paper.	Discussed in detail in the paper.
	Sub-system 2: Geolocation Prediction	**Source:** Twitter **Samples Count:** ~ 577K **Geo-Area:** USA, UK, India, Australia, Canada	**Input:** Prepared dataset (the user profile description, user screen name, user profile location, tweet text, and calculated tweet sentiment) **Output:** Tweet location for the most 5 hazardous geolocations worldwide	Classical ML Algorithms such as (MNB/DT/ET/RF/ Ada/GBM) with different vectorization, DL Algorithms such as (MLP, CNN, LSTM, BiLSTM) with different word embedding	Accuracy: ET, 80.8% Bi-LSTM+RoBERTa, 96%	Discussed in detail in the paper.	Discussed in detail in the paper.
	Sub-system 3: Public Awareness Quantification	**Source:** Twitter **Samples Count:** ~ 577K **Geo-Area:** USA, UK, India, Australia, Canada	**Input:** Tweet Volume, Tweet Sentiment, Tweet Location Outbreak formal reporting **Output:** Correlation coefficients and lag/lead time between sentiment-based Tweets activities time series and no. of infected cases time series	Windowed Time Lagged Cross-Correlation	Correlation coefficients: 0.91- 0.96, Time Period: A temporal lag between the rise in the number of COVID reporting tweets and officially reported positive cases varies between 13 to 26 days.	Discussed in detail in the paper.	Discussed in detail in the paper.

TABLE 3. **Dataset description.**

Dataset Source	Twitter, IEEE data port [85, 86]
Total Number of Instances	577K Tweet
Number of Attributes	24
Selected Features Information	user profile description user screen name user profile location tweet text tweet sentiment
Output	Tweet's sentiment, Tweet's location, Time of lag or lead.
Classes Distribution	Unbalanced

improve the awareness prospect of infectious disease as a proactive info-surveillance framework. One of the objectives of this research is to investigate how population sentiments can be correlated with the emergence of an infectious outbreak and how this can be referenced to certain geolocations worldwide. So, in accordance with this primary goal, this research aims to answer the following questions:

Research Question (1): What is the potential expediency of employing social network data and spatiotemporal social network analysis to proactively control any infectious disease outbreak spread?

Research Question (2): How could the employment of spatiotemporal social network analysis in studying the progression of an epidemic outbreak overcome the challenges that healthcare authorities encounter, such as governments' limited data collection resources, diversity in cultural and educational backgrounds, and inconsistencies in reporting from various sectors?

Research Question (3): To what extent does the extracted linguistic features vector associated with social network data affect the performance measures of classification algorithms employed either in predicting the sentiment associated with a text or in predicting the geolocation from which the text is originated?

Research Question (4): What are the expected improvements upon applying modern text vectorization algorithms such as pre-trained neural networks and transformer-based word embedding vectorization compared to classical vectorization techniques when used with social network textual data?

Research Question (5): Does the employment of neural network-based classification models affect the resulting accuracy compared to the employment of state-of-the-art machine learning-based models using the same data corpus?

Research Question (6): How would classical correlation and time-lagged cross-correlation algorithms quantify the synchrony between time-series data, and how can these algorithms be employed in developing an early warning and surveillance framework for any infectious disease outbreak?

B. PROPOSED FRAMEWORK

This study utilizes social media data obtained from Twitter and investigates the potential of capturing public sentiment to promote spatiotemporal early warning systems. This study retroactively investigates the early stages of the COVID-19 disease outbreak worldwide to detect the most hazardous geolocation. Numerous technologies and algorithms have been employed for the research problem under consideration to achieve the highest accuracy and best performance. As such, a proposed framework that consists of three major inherently interlinked and cooperative subsystems is proposed. The first subsystem consists of data collection and integration mechanisms and APIs, data preprocessing, and hybrid sentiment analysis tools that identify tweet sentiment taxonomies and perform quantitative analysis to estimate public awareness. The second subsystem comprises the feature extraction unit that identifies, selects, embeds, and balances feature vectors and the classifier fitting and training unit. This subsystem is designed to capture the most effective linguistic feature combinations with more spatial evidence by employing various word embedding algorithms with diverse neural network-based models. The last subsystem is the modeling and situational awareness evaluation subsystem based on the machine and deep learning techniques and used for predicting the geolocation of most affected countries using online social network data. The public situational awareness evaluation is designed based on various correlation quantifying approaches and developed to measure the temporal associations (synchrony) between public pandemic relevant online social network activities and pandemic officially announced infection counts in the most hazardous geolocations. So, collectively, the proposed framework includes three major components that will be employed in the previously mentioned subsystems, the first is the generic classifier training and fitting algorithm, the second is the predictor (classifier), and the third is the correlator. The classifier training and fitting process is divided into several serial sub-processes, commencing with collecting social network data, moving on to data preparation and preprocessing, then reaching the calculation of sentiments associated with the scraped data. The preprocessed text along with its predicted sentiment, are combined for the feature vector extraction. These features can be divided into training and validation datasets used to test numerous machine learning-based and neural network-based classification models. This process concludes with the selection of the best classification model based on the performance evaluation sub-process, as depicted in the block diagram presented in Fig. 1, which illustrates the main building blocks of this main functional process in addition to clarifying the input and output of each block and detailed in Fig. 2 below, where the internal functions of each sub-process are declared. At that juncture, the predictor (classifier) process is enabled to employ the selected best classifier and the features vector from the previous process to classify any new dataset. Finally, the correlator process employs different cross-correlation algorithms to quantify synchrony between the predicted sentiment of text data related to a certain epidemic outbreak and the official reporting of daily active infections for the most hazardous geolocations extracted from the previous classifier process. The predictor (classifier) and the correlator processes are illustrated as block diagrams in Fig. 3 and Fig. 4 and detailed in Fig. 5, Fig. 6, and Fig. 7, below where the internal functions of each sub-process are dismantled.

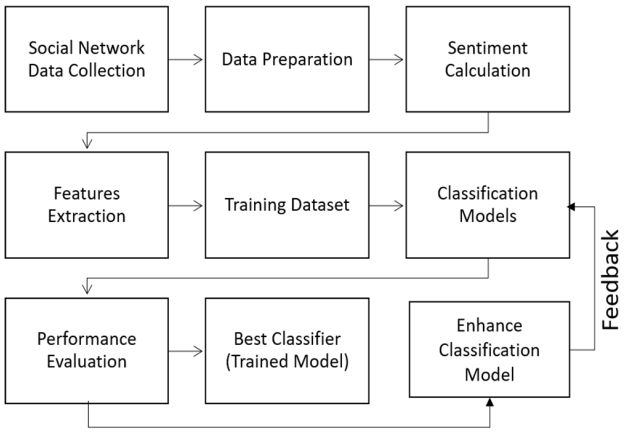

FIGURE 1. Classifier training and fitting process block diagram.

A deeper explanation of the mentioned sub-process and their functions are presented in an upcoming section. Moreover, the main contributions of this research are elaborated as follows:

1. Generation of a dynamic dataset comprising real-time scraped and archived data, with reasonable data size and features.
2. The proposed framework is based on textual data and metadata that are voluntarily publicly shared via social network platforms while preserving all of the users' privacy rights – such that only data permitted by the user is available to the public community and, in turn, to the research community. This enables the proposed framework's reproducibility. Furthermore, the project's GitHub repository contains all implementation files, codes, libraries, environment setup and requirements, dataset, and results are accessible via: https://github.com/nohagamal001/Sentiment-Based-Spatiotemporal-Awareness-Framework-for-Pandemics
3. For real-time and archived data collection of English geotagged tweets, the proposed framework provides its data scraping and hydrating APIs, which are designed to use a set of corona-virus-related keywords. It was necessary

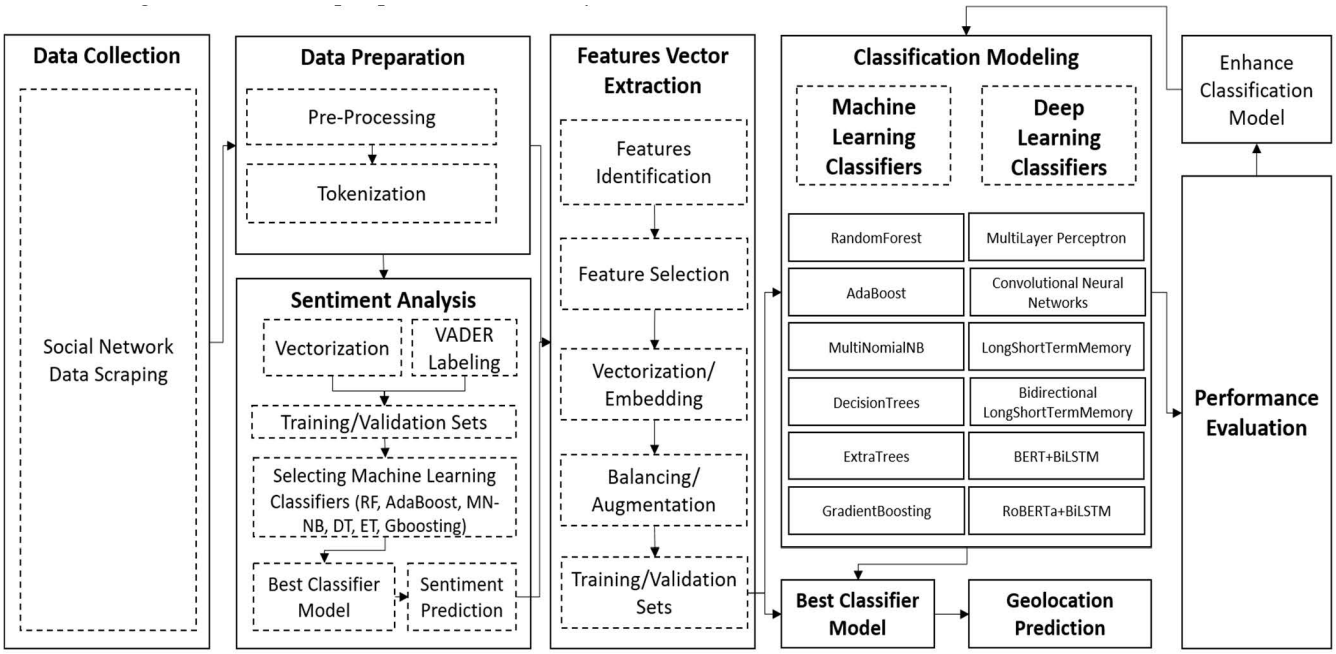

FIGURE 2. Proposed framework, training/validation phase.

to collect tweets from all over the world so that our framework could detect the most hazardous geolocations according to the implemented processes and also allow us to evaluate the performance of our model in distinguishing between different variants of a single language. As a result, we tested the framework's reliability in five different regions (USA/Canada/UK/Australia/India).

4. This framework incorporates a tweet sentiment prediction as an additional feature to the tweet text, user screen name, and profile description in order to predict the tweet's geolocation (place). Although this new feature hasn't been used in similar research, our literature and knowledge have been shown to improve tweet classification-based location prediction with both classical machine learning and neural network algorithms. Additionally, the new feature provided a deeper level of insight into the collected data, allowing researchers to pinpoint the most vulnerable and infection-prone areas.
5. In our proposed framework, depending on textual inputs, it is very important to recognize the word sense or meaning as used in a text originating from a specific location for the same language, as a different geographical region may give a different meaning to a word (regional context). So, in addition to employing and testing conventional machine learning-based classification models, we have tailored and evaluated many neural network-based algorithms, such as Feed-Forward Multi-Layer Perceptron, Recurrent Neural Network (LSTM, Bi-LSTM), and Convolutional Neural Network, empowered by self-embedding, pre-trained static and dynamic word embedding such as "Word2vec, FastText, Glove" and "BERT and RoBERTa" respectively. These tailored models were successful by nature in capturing the regional context of a word.
6. To broaden the scope of our evaluation and identify the best algorithms for social network data, we selected pre-trained word embedding algorithms that allow our proposed framework to evaluate various classical and up-to-date embedding algorithms developed from 2010 to 2020.
7. Using Bi-directional Neural Networks allowed us to predict the geolocation of a tweet based on the spatial indications of words in relation to their context and position (index) in the text. This capability was recognized when the Bi-directional Recurrent Neural Network empowered by Bi-directional pre-trained word embedding outperformed other implemented Neural Network models like MLP and CNN. When compared to state-of-the-art approaches presented in the literature, the proposed framework with the effective selection of different algorithms, hyper-parameters, and configurations resulted in noticeably better geolocation prediction accuracy.
8. As our implementation deals with linguistic features rather than relying on lexicons, the same high standard performance is guaranteed when our framework is used to process different corpora written in English. Furthermore, it can be extended to process non-English corpora by using appropriate word embedding algorithms capable of dealing with new language linguistic features, such as using Ara-BERT to process Arabic corpora within our proposed implementation.
9. The proposed framework presents an evaluation for the implementation of time-lagged cross-correlation (TLCC) and windowed-time-lagged cross-correlation (WTLCC)

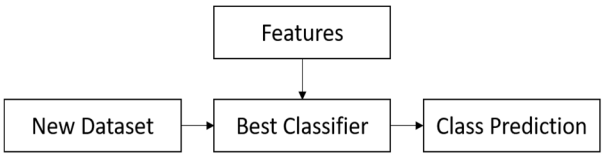

FIGURE 3. The predictor (Classifier) process block diagram.

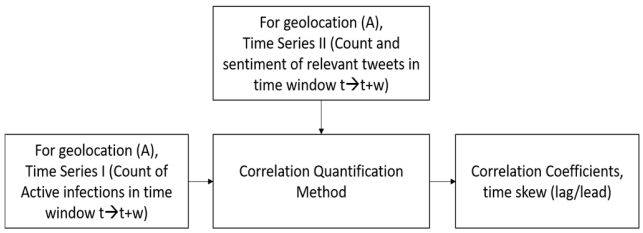

FIGURE 4. The correlator process block diagram.

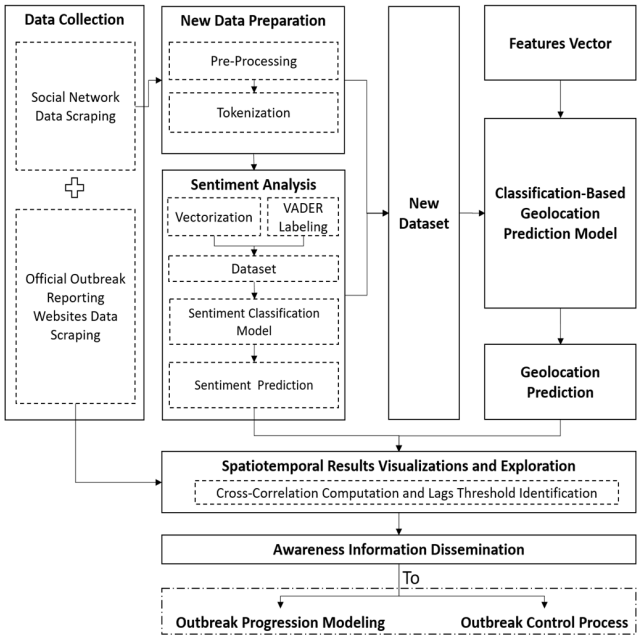

FIGURE 5. Proposed framework, evaluation phase, (the processes indicated in the dash-dot block is out of this study's scope, as these will be included in our future work studies).

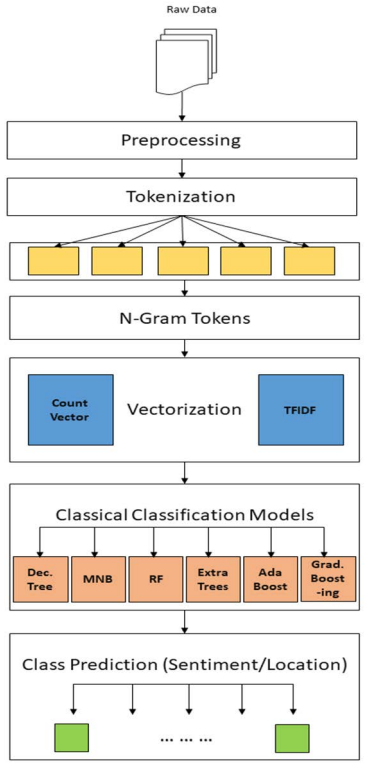

FIGURE 6. Illustration for the proposed machine learning classification model.

algorithms to quantify the synchrony between the public interaction volumes and sentiments, and the official reports of the COVID-19 infection active cases in geolocations under investigation. The WTLCC was decided to be employed because of its capability to deal with the nature of the outbreak in waves. To the best of our knowledge, no prior studies have employed WTLCC to solve this type of research problem.

10. As our implementation deals with linguistic features rather than relying on lexicons, the same high standard performance is guaranteed when our framework is used to process different corpora written in English. Furthermore, it can be extended to process non-English corpora by using appropriate word embedding algorithms capable of dealing with new language linguistic features, such as using Ara-BERT to process Arabic corpora within our proposed implementation.
11. The proposed framework employed a multi-staged process to measure the potential expediency of using social network data along with official reports about an infectious disease outbreak. By implementing and employing many efficient algorithms and methods within the proposed framework, starting from real-time topic-related data scraping, passing by evaluating the public sentiment in regards to the topic during different phases and time periods of its emergence, which was employed amongst other features to predict the most hazardous geolocations, then finally reaching to quantifying the synchrony between the public interaction volumes and sentiments, and the official reports about the topic under investigation (COVID-19 outbreak) in certain geolocations that can be marked as the most jeopardized.
12. The proposed framework evaluates the performance of numerous algorithm combinations and integrations from the perspectives of accuracy and required computational resources in order to present a comprehensive analysis to the research community that will assist researchers in deciding which experiments to rebuild or improve based on their research problem.
13. To the best of our knowledge, no prior studies have implemented such a multi-staged framework that measures the spatiotemporal social network potentials to aid in the mitigation of infectious disease spread in its early stages

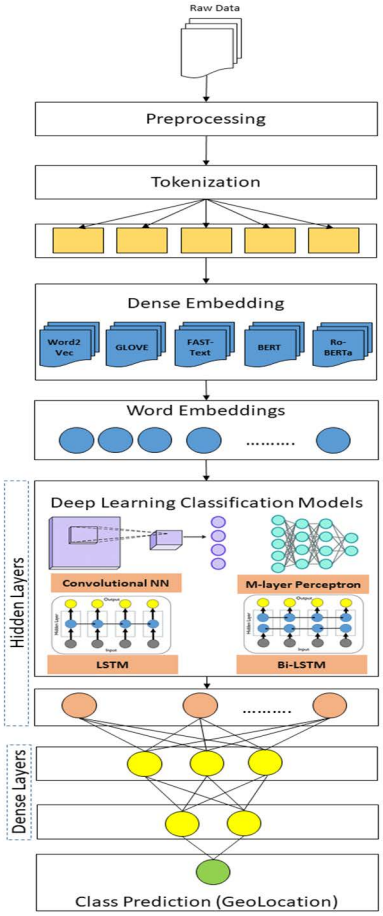

FIGURE 7. Illustration for the proposed deep learning classification model.

when insufficient data is available in the manner we have proposed.

Process 1:

Process 1(a): Scraping Real-time tweets by keywords

1. Log in to the Twitter developer section
2. Create a new Twitter application with the appropriate details
3. Details of the new app are approved, and the consumer key and consumer secret are available on the application portal
4. Generate application access tokens
5. Import tweepy # python-based library for scraping from Twitter
6. Setup the application authentication:
   Auth=tweepy.OAuthHandler(Consumer_Key, Consumer_Secret)
   Auth.set_access_tokens=(Access_Key, Access_Secret)
7. Call the new application using tweepy:
   api= tweepy.API(Auth)
8. define get_Real_time_tweets(keyword, count = 100)
   # call twitter api to fetch tweets
   keyword="keyword0"
   q1=str(keyword+" keyword1")
   q2=str(keyword +" keyword2")
   q3=str(keyword +" keyword3") #can be repeated for unlimited number of keywords and keywords combinations.
   Tweets = api.search(q1, count = count)+ api.search(q2, count = count)+ api.search(q2, count = count)
   Select the required parameters to generate the implememtation dataset:
   Tweets_set=Tweets(tweet_id, tweet_text,user_screen_name,user_profile_description, user_location)
9. Save scraped tweets to .txt or. CSV file for further processing

Process 1(b): hydrating archived tweets by tweet_id

1. Repeat Algorithm 1(a) from step1 to step 6
7. Call the new application using Twarc:
   Tweets= Twarc(Consumer_Key, Consumer_Secret, Access_Key, Access_Secret)
8. define get_Archived_ tweets(tweet_id, count = 100)
   # call twitter api to fetch tweets
   For tweet in Tweets.hydrate(open('list_of_tweets_ids')) do:
   Select the required parameters to generate the implememtation dataset:
   Tweet_id=tweet["id"], Tweet_text=tweet["full_text"],
   Tweet_user_name=tweet["user_screen_name"],
   Tweet_user_location=tweet["user_location"]
   Tweets_set=Tweets(tweet_id, tweet_text,user_screen_name,user_profile_description, user_location)
9. Save scraped tweets to a .txt or CSV file for further processing

FIGURE 8. Pseudocode of process 1, process 1(a): Scraping real-time tweets by keywords and process 1(b): hydrating archived tweets by tweet_id.

C. MATERIALS AND METHODS

1) DATA COLLECTION

Twitter data could be extracted either by using live (real-time) scraping or by hydrating older tweets using their tweet IDs. In either case, a developer Twitter account should be created, and an application should be signed up. Both ways (real-time and hydrating) were used to be able to collect the most suitable dataset for our proposed framework as per the following pseudocode illustrated in Fig. 8. For 18 months, from March 2020 to September 2021, the Twitter streaming API was used to gather real-time geo-tagged tweets from around the world. For early COVID-19-related tweets, we have hydrated the GeoCOV19 Tweets Dataset [85]. During the mentioned period of time, the dataset should include 427,447 English tweets, with global coverage as announced on [86]. This dataset includes the IDs and sentiment scores for these geotagged tweets about the COVID-19 epidemic. Only the tweet IDs are available on the cited website, per Twitter's content redistribution policy. By hydrating these IDs, we reconstructed a dataset of around 349,760 unique tweets. In addition, around 227,500 other tweets were directly scraped from Twitter's real-time streaming.

2) TWEETS DATA CLEANING AND PREPROCESSING

Preprocessing text is used to remove noisy or inconsistent data, these include eliminating punctuation, special characters, numerals, and words that have no significance in the context of the text. The accurate data preparation and preprocessing will definitely lead to better feature space selection and so much accurate classifications and predictions. Main cleaning and preprocessing steps were tackled using NLTK2, and TextBlob libraries are listed as shown in the pseudocode illustrated in Fig. 9. Many other functions were employed to finalize the text cleaning and preprocessing of tweets, such as; Replacing all emoticons with their polarity of meaning, using the dictionary of emoticons, replacing all acronyms with their translations, using the acronym dictionary, using the negation dictionary to substitute negation terms with "non.", and Stemming, that is is a rule-based process of stripping the suffixes ("ly," "ing," "es," "s," "ed,"

Process 2: Tweets dataset cleaning and preprocessing

```
1  Tweets_set=Tweets(tweet_id,
   tweet_text,user_screen_name,user_profile_description, user_location)
2  def clean_tweets(tweet_text): #using python regular expression module
   "re" and substitue function "sub"
    #Remove hyperlinks:
      tweet_text=re.sub('https?:\/\/.*\/\w*','', tweet_text)
    # Remove citations: tweet_text = re.sub('\@\w*', '', tweet_text)
    # Remove tickers: tweet_text = re.sub('\$\w*', '', tweet_text)
    # Remove punctuation:
      tweet_text = re.sub('[' + string.punctuation + ']+', '', tweet_text)
    # Remove quotes:   tweet_text = re.sub('\&*[amp]*\;|gt+', '', tweet_text)
    # Remove RT:   tweet_text = re.sub('RT', '', tweet_text)
    # Remove linebreak, tab, return:
      tweet_text = re.sub('[\n\t\r]+', ' ', tweet_text)
    # Remove via with blank:   tweet_text = re.sub('via+\s', '', tweet_text)
    # Remove whitespace:   tweet_text = re.sub('\s+\s+', ' ', tweet_text)
    # Remove HashTags:
      tweet_text = re.sub('\#+[\w_]+[\w\'_\-]*[\w_]+', ' ', tweet_text)
3  For tweet in Tweets_set:
   If (tweet_id is Not None) and (tweet_text is Not None) and
   (user_screen_name is Not None) and (user_profile_description is Not None)
   and (user_location is Not None) Do:
   Cleaned_tweets= clean_tweets(Tweets_set)
```

FIGURE 9. Pseudocode of Process 2, tweets dataset cleaning and preprocessing.

etc.) from a word. The Porter Stemmer from the NLTK library was used to resolve this issue. Finally, unigram tokenization is applied to all tweets. Tokenization is the process of splitting a string of text into tokens, where tokens are individual terms or words.

3) DATA BALANCING AND AUGMENTATION

The used dataset showed some distribution imbalance which might cause over-fitting or under-fitting of certain classes, so, an additional step was added after data cleaning is to use Data Augmentation. To increase the amount of data by adding newly generated synthetic data from current data, these act as a regularizer and helps reduce overfitting when training a machine or deep learning model. Additionally, a hybrid strategy of combined over-sampling and under-sampling techniques was used to adjust the class distribution of our dataset.

4) DATASET DESCRIPTION

After applying the basic data cleaning process, the dataset enclosed more than 577K unique geotagged tweets, with many collected features. There is a lot of information associated with tweets, including the user screen name, userID, following date and time, count of followers, tweet's text, user's device type, such as Android or iOS, in addition to, geolocation, user biography, count of user mentions, and the count of re-tweets. So, amongst these features, the user profile description, user screen name, user profile location, tweet text, and calculated tweet sentiment have been selected to predict the geolocation ''place'' affected by the pandemic.

5) DATA VECTORIZATION, TRANSFORMATION, AND FEATURES EXTRACTION

Most text-based classification techniques use machine learning and deep learning models. The input used is mostly unigram or n-gram tokens, or what can be said as a ''bag of words'' (BoW) [87]. However, BoW tokenization is not sufficient in complex classification problems such as location prediction. Hence, more specific features are needed to be extracted for a more accurate classification process. Word embedding [88]–[93] is now one of the most successful applications of unsupervised learning since it does not require expensive annotation. Word embeddings are represented in such a way that comparable words are encoded in the same way. Word embedding helps with feature extraction and natural language processing applications such as text classification, topic clustering, and sentiment classification. Traditional or frequency word embedding, static word embedding, and contextualized dynamic word embedding are the three types of word embedding techniques. The count vector, Term Frequency – Inverse Document Frequency (TF-IDF) [88], and co-occurrence are the three types of frequency-based embedding. Word2Vec [89], GloVe [90], and FastText [91] are the three types of static word embedding. BERT [92] and RoBERTa [93] are two types of contextualized word embedding. For the purpose of our study in the proposed framework, many feature selection techniques have been tackled, as shown below, in order to improve the model's performance accuracy and reliability. The CountVectorizer is used to convert a given text into a vector of numeric values based on the frequency (count) of each word in the text [94].

This is advantageous if we have a collection of such texts and wish to transform each word in each text into a vector. Thus, if a train set has ten tweets, those ten tweets are tokenized to produce the bag. TF-IDF is an additional technique based on the frequency method, but it differs from the CountVectorizer approach in that it considers the recurrence of a word across the entire corpus rather than just in a single document (or tweet). The TF-IDF algorithm penalizes popular terms by assigning them lower weights while emphasizing words uncommon throughout the dataset but appearing in large numbers in a few documents [95].

The BoW and TF/IDF vectorization implementation are shown in the pseudocode illustrated in Fig. 10. The second model is a time series approach based on word embedding. In which each word is represented by its vector in this instance. Word embedding can be implemented by building a problem-customized self-embedding or using pre-trained embedding matrices. For self-embedding, each text sample will be passed through an embedding layer that calculates the proximity between features to be able to assign a location to the word in the n-dimensional embedding space.

Or, we may use pre-trained word embeddings, such as Google's word2vec [89], Stanford's GloVe Text [90], and FastText [91]. These are trained using massive corpora, including billions of instances and words. When a model is trained on a large text corpus with the assistance of a high-performance cluster. Later, it may be fine-tuned for a given purpose in considerably less time. Additional layers might be added to the model during the fine-tuning stage for particular tasks distinct from those it was first trained. These word

Process 3: Tweets Vectorization (BoW) Vs. (TF/IDF)	
1	Cleaned_tweets= clean_tweets(Tweets_set) remove the extremely common words: stop_words="english" 'n' most frequent words: max_features= 1000 vector is using word level or char level: analyzer="word" n-gram or a string of words in a row: ngram =(1,1) #default
2	**BoW Vectors:** vectorizer = CountVectorizer(max_features, stop_words, analyzer, ngram_range) BoW= vectorizer.fit_transform(Cleaned_tweets['tweet_text'])
3	**TF/IDF Vectors:** vectorizer=TfidfVectorizer(max_features,stop_words,analyzer,ngram) TF_IDF = vectorizer.fit_transform(Cleaned_tweets['tweet_text'])

FIGURE 10. Pseudocode of process 3, tweets vectorization (BoW) vs. (TF/IDF).

embedding algorithms will be discussed in further detail in a subsequent subsection.

D. SENTIMENT ANALYSIS

Because we are trying to anticipate where a particular event is emerging, like the COVID-19 epidemic, will have the most impact, understanding public sentiment is critical to the success of our research. The proposed location prediction approach takes into account sentiment classifications as well as other textual inputs. Calculating sentiment scores for all tweets included in our dataset may thus be seen as a crucial job that can lead to precise location predictions being performed. Since public sentiment classification emerged, several new approaches have developed, such as machine and deep learning techniques for automated, rapid, and accurate classification of sentiments. Natural Language Processing (NLP) addresses this issue by combining linguistics and computer science to convert text to a format that computers can comprehend [96]. While certain natural language processing models are more sophisticated than others, most sentiment classification systems use one of three algorithms: Rule-Based Sentiment Classification (or Lexicon-Based), Automated Sentiment Classification (or Machine Learning-Based), and Hybrid Sentiment Classification [33], [34]. For sentiment classification tasks, rule-based systems depend on a lexicon (dictionary or corpus-based), which is a collection of positive and negative words. When given a piece of text, the model counts positive and negative words and assigns sentiment to each. A sentence is labeled ''positive'' if it includes more positive than negative terms. This strategy, however, has a few drawbacks. It is unable to detect words that do not occur in the lexicon and disassociates words from their context units, making it difficult to distinguish sarcasm or irony [97]. Finally, rule-based systems might be challenging to scale and improve since new terms added to the lexicon may alter prior findings. Automated systems make use of machine learning techniques to anticipate sentiment based on prior observations. A training dataset and its associated tags or sentiment classes are essential to using this AI method. During the training phase, the model converts text input to vectors and determines a pattern to connect each vector with one of the pre-defined tags (''Positive,'' ''Negative,'' or ''Neutral''). As a sufficient quantity of relevant data is supplied into the system, automated systems may begin to form their predictions to classify previously unknown data. It is worth mentioning that the larger the training dataset, the more accurate the machine learning models are. Hybrid systems combine techniques based on rules and machine learning. To commence, the model trains on a sequence of tagged instances to identify sentiment. The findings are then compared against a lexicon to ensure they are accurate. The objective is to get the optimal result while avoiding the drawbacks inherent in each particular strategy. In our proposed scheme, we have implemented a hybrid sentiment classifier to tag the tweets dataset with the appropriate sentiment, as illustrated in the pseudocode presented in Fig. 11.

E. ARTIFICIAL INTELLIGENCE-BASED SOCIAL NETWORK ANALYSIS AND CLASSIFICATION METHODS

1) MACHINE LEARNING-BASED CLASSIFICATION METHODS

It is important to evaluate many models regardless of their theoretical performance when using machine learning since the accuracy is based on the training dataset. For different reasons, such as linear separation or the burden of dimensionality, a few methods (SVM, Naïve Bayes, multinomial regressions) are commonly used for text classification tasks [98]. However, we have decided to implement many other algorithms such as MultinomialNB, DecisionTreeClassifier, as examples of single classifier algorithms, and ExtraTreesClassifier, RandomForestClassifier, AdaBoostClassifier, and GradientBoostingClassifier, as examples of ensemble learning classifier algorithms, to examine their performance in either sentiment classification or location prediction such as illustrated pseudocode of process 4, shown in Fig. 11, by using different input features and output labels in either case. In the following subsection, we describe the employed customization on conventional machine learning models and their main parameter specifications.

a: MULTINOMIAL NAÏVE BAYES

Naïve Bayes statistical methods are some of the most often employed in text classification and analysis. Experiments may achieve excellent results even if the dataset is small and computing resources are limited by using Multinomial Naïve Bayes (MNB), one of that family members. It is possible to determine the conditional probability of two occurrences based on the probabilities of each event using Bayes's Theorem. Our method calculates each class's likelihood for a given text, then outputs the class with the greatest probability [99]. As a result, each vector represents a piece of must-have text information on the probability of certain words appearing in texts of a particular category for the algorithm to calculate the likelihood that the text belongs to the category. In our experiment, we have implemented the MNB classifier with Laplace smoothing and prior probabilities fitting set to True.

b: DECISION TREE CLASSIFIER

A decision Tree (DT) is a strategy for classifying decisions based on the use of branches to represent each potential

Process 4: Classification model using Machine Learning
#Ex. Tweets Sentiment Prediction (Hybrid Method)

```
1  from vaderSentiment.vaderSentiment module: import
   SentimentIntensityAnalyzer class.
2  Calculate Rule-based (Lexicon) Sentiment using VADER
   def sent_scores(sentence):
   # Create a Sentiment Analyzer object.
     sa_obj = SentimentIntensityAnalyzer()
   # object gives a sentiment dictionary. that contains pos, neg, neu, and comp
   scores.
   For Tweet in TF_IDF(Tweets_set) :
   sentiment_dictionary = sa_obj.polarity_scores(sentence)
   # decide sentiment as positive, negative and neutral
   if sentiment_dictionary['comp']>=0.05: tweet.sentiment= "Pos"
   elif sentiment_dictionary['comp'] <= - 0.05 :  tweet.sentiment="Neg"
   else : tweet.sentiment="Neu"
3  Update Sentiment field for all Tweets:
   for Tweet in Tweets_set: Tweet['sentiment]= tweet.sentiment
4  Calculate Machine Learning-based (Automated) Sentiment
   Classifiers = [ RandomForestClassifier(n_estimators=200, criterion='gini',
   max_depth=50, max_features='auto'),
   MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True),
   tree.DecisionTreeClassifier(),
   ensemble.ExtraTreesClassifier(n_estimators=100,    max_features=  50,
   criterion= 'entropy'),
   ensemble.GradientBoostingClassifier(criterion='friedman_mse',
   learning_rate=0.001,n_estimators=50, random_state=None, verbose = 0) ]
   for classifier in Classifiers:
   fit = classifier.fit(train_features, Tweet['sentiment'])
    pred = fit.predict(test_features)
5  Evaluate different Classifiers Performance by computing the accuracy
   accuracy = accuracy_score(pred, Tweet ['sentiment'])
```

FIGURE 11. Pseudocode of process 4, classification model using machine learning.

outcome. There are three types of nodes in DT: the ''root node,'' the ''internal node,'' and the ''leaf node.'' An initial attribute or the uppermost decision node in a tree, the root node, is the best predictor for a tree. On the other hand, internal nodes have at least one incoming and outgoing edge. A decision or categorization is indicated by a leaf node that has no outgoing edges. Decision Trees (DT) learn from data to approximate the sine curve using an IF-THEN rule set [100]. As the tree becomes more complicated, so does the model's ability to accurately predict future outcomes. When DT is used to classify decision-making data, it's easier to access and analyze and requires less computing power while also being able to show the link between dependent and independent variables while also being computationally low-end. DT may be used as a basic framework that sets a group of rules, which is used for decision-making to categorize a document based on its content into a certain class. In our experiment, we are using a standard Skicit-Learn Decision Tree.

c: ENSEMBLE LEARNING

Ensemble learning – approaches that incorporate numerous classifiers and integrate their findings – has sparked a lot of interest in machine learning research. Given the same amount of training information, it is generally anticipated that the performance of a group of numerous weak classifiers outperforms that of a single classifier. Boosting, bagging, and, more recently, Random Forests are well-known ensemble approaches. By successively reweighing the examples in the training set, the boosting strategy generates various base learners. At the onset, all instances are given equal weights. In the following round, each case misclassified by the previous base learner will be given a higher weight to attempt more accurate classification. The bagging approach (bootstrap aggregation) draws various training subsets at random from the total training set. Each training subset is delivered to basic learners as input. A majority vote is used to aggregate all retrieved learners. While bagging may produce classifiers in parallel, boosting produces them sequentially. One famous example of boosting ensemble is gradient boosting-based classifiers that combine weak learning models to create a powerful prediction model. Gradient boosting methods can successfully handle complicated unstructured social media data. Gradient Boosting Machines (GBM) have shown state-of-the-art performance on a variety of conventional classification benchmarks [101]. Gradient Boosting machines' main goal is to minimize the loss function, which is analogous to gradient descent methods in neural networks. New weak learners are added to the model iteratively, and the weights of previous learners are locked in place, leaving unmodified samples for the new layers. Multi-class classification and regression issues may be solved with GBM. A GradientBoostingClassifier was implemented considering 100 estimators and an error-minimizing criterion as ''friedman_mse''. Another example of a boosting ensemble is Adaboost. In Adaboost, the greater the number of misclassifications, the greater the number of weights allocated to the misclassified to improve or enhance the algorithm's performance to provide a more accurate forecast. AdaBoostClassifier is implemented using 100 estimators. Random Forest, as a different ensemble approach, constructs many decision trees that are used to classify a new entity by majority vote. As a result, each node in the decision tree uses a randomly selected subset of the original collection's features. Furthermore, each tree, similar to bagging, uses a different bootstrap sample data. Bagging is often more accurate than a single classifier, yet in some circumstances, it may perform significantly less accurately than boosting. Boosting, on the other hand, may produce ensembles with lower accuracy when compared to a single classifier's accuracy [102]. Moreover, boosting may overfit noisy datasets, resulting in poor accuracy. Random Forests, contrarywise, are more resistant to noise than boosting and perform as well as, if not better than, boosting, in addition to its resilience to overfitting. In general, the user determines the number of trees constructed in the forest by trial and error. When the number of trees increases, more computing power is needed, with no noticeable performance advantage. A RandomForestClassifier was implemented considering 200 estimators (the number of trees in the forest), and an error-minimizing criterion as ''Gini'' impurity. We are constructing numerous decision trees as an illustration of ensemble classification techniques. RandomForest determines the best split for converting the parent node into the two most homogenous child nodes. As the name implies (Extremely Randomized Trees), an ExtraTreesClassifier chooses a random split to divide the parent node into two random child nodes. An ExtraTree Classifier was built using 100 estimators (forest tree count) and an error minimization

TABLE 4. Neural network hyper parameters.

Parameter	Value
Sequence length	30
Number of classes	5
Vocabulary Size	40K
Embedded Vector Size	300
Tensor Dimensions	64 × 30 × 300
Embedded matrix	40000 × 300
Epochs	10
Learning rate	10^{-4}
Optimizer	Adam
Dropout rate	0.1

criteria known as ''entropy.'' Random forest models mitigate overfitting by incorporating randomness via the construction of numerous trees (n-estimators), the use of bootstrapped samples, and the splitting of nodes based on the optimal split among a casual subset of the features picked at each node. ExtraTrees is like Random Forest in that it generates numerous trees and divides nodes using random subsets of features, but with two significant differences: it does not require bootstrap observations, and nodes are divided using random splits rather than optimal splits. Thus, randomization in Extra Trees is not derived from bootstrapping the data but rather from random splits of all observations.

2) DEEP LEARNING-BASED CLASSIFICATION METHODS

Natural Language Processing is a critical technology in this age of information and data. With the immense presence of word embedding, neural network models have attained very high-performance metrics across a wide range of NLP applications. The essential features and characteristics used in implementing these models are explained in the upcoming subsections. In addition to presenting the detailed layers architecture for CNN, LSTM, and Bi-LSTM with Self-Embedding Layer, as shown in Fig. 12, we demonstrate the complete process for implementing such networks as depicted in the pseudocode presented in Fig. 13. The following subsection discusses the customization employed on neural network learning models and the main parameter specifications. The deployed parameters were selected with reference to the literature and experimentally optimized for the best performance. Additionally, Adam optimizer was employed for further performance enhancements. Part of the deployed parameters is presented in Table 4 below, while the remaining parameters are mentioned within each algorithm described in the following subsections.

a: MULTI-LAYER PERCEPTRON NEURAL NETWORK

A multi-layer perceptron (MLP) is a kind of forward-feed artificial neural network (ANN) [49]. MLP is a word that is sometimes used generically to refer to any feedforward ANN and other times solely to networks built of many layers of perceptrons (with threshold activation). An MLP is composed of at least three layers of nodes: an input layer, a hidden

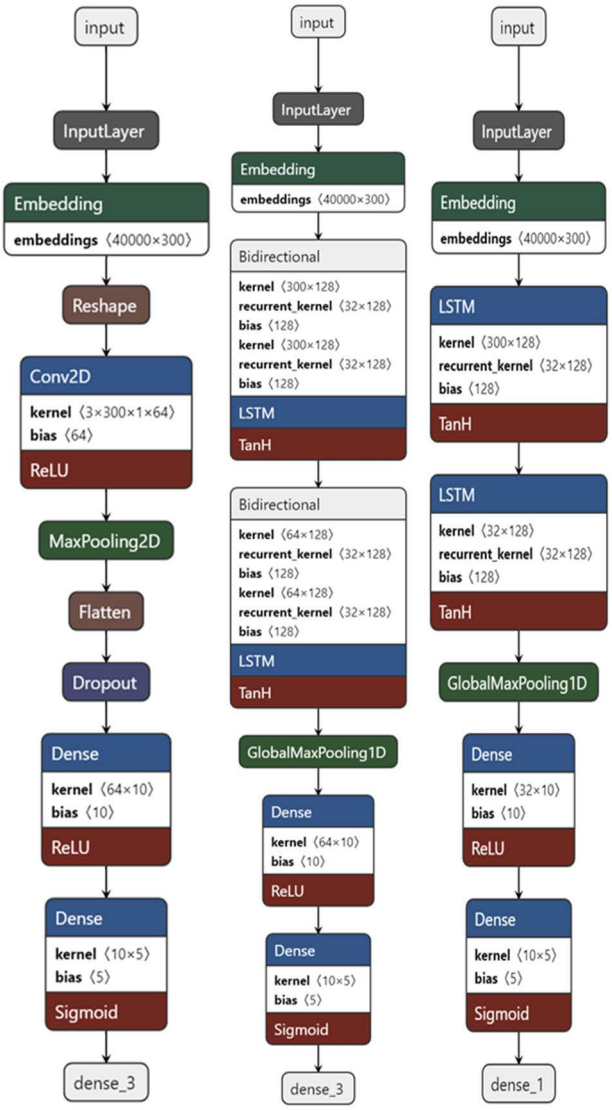

FIGURE 12. Proposed models architectures for CNN, Bi-LSTM, and LSTM with embedding layer.

layer, and an output layer, in which each node is a neuron with a nonlinear activation function except the input nodes [77]. The output of the embedded layers will be flattened to one dimension, and a dense hidden layer of 256 units with a rectifier activation function will be used. MLP is trained using a supervised learning approach known as backpropagation. A multi-layer perceptron (MLP) with two hidden layers is employed in our proposed method, as indicated in the pseudocode of process 5, shown in Fig. 13, which is presented as an example of the common process we have employed to build any of the proposed neural networks with any of the suggested word embedding models.

b: CONVOLUTIONAL NEURAL NETWORKS

Recent developments in Natural Language Processing (NLP) heavily rely on neural network models. Convolutional Neural Network (CNN) solutions are used to solve NLP problems,

	Process 5: Tweets Location Prediction (using NN and word embeddings)
1	**Set Embedding Parameters values:** {'embed_dim': 300, 'max_len': 200, 'pos_embed_dim': 50, 'trainable_param': True, 'use_pos': True, 'vocab_size': 40000, 'window': 1, 'workers':4 } **Set Neural Network model_paramseters values:** {'batch_size':64,'dropout':0.1,'epochs':10, 'hidden_dims':128,'loss':['binary_crossentropy','categorical_crossentropy', 'sparse_categorical_crossentropy'], 'metrics': ['accuracy'], 'nb_filter': 64, 'optimizer': 'adam', 'recurrent_dropout': 0.1} **Set Pre-trained Embedding_options:** {0 : Word2vec, 1 : Fasttext 2018, 2 : GloVe, 3 : Word2vec+PoS, 4 : Fasttext 2018+PoS, 5 : GloVe+PoS} **Set Neural Network Type_Options:** {0: simple 2layersMLP,1: lstm,2: bilstm, 3: cnn}
2	**Define load_data:** Args: vocab_size = {int} the size of the vocabulary, max_len = {int} the maximum length of input considered for padding Returns: X_train = tokenized train data , X_test = tokenized test data
3	**Define prepare_data_for_word_vectors:** Args: X_train = tokenized train data, X_test = tokenized test data Returns: sentences = {list} sentences containing words as tokens, word_index = {dict} word and its indexes in whole of corpus
4	**Define class Embed:** __init__(self,vocab_size,embed_dim,pos_output_dim,max_len,pos_trainable_par am): **def embed_sentences** (self,word_index,model,trainable_par,X_train_pad): return input_seq,embed_seq **def tag_pos**(self,sentences): return np.array(X_pos_encoded) **def embed_pos**(self,X_pos_arr): return input_seq_pos,embed_seq_pos **def pos_model_build**(input_seq, input_seq_pos, embed_seq, embed_seq_pos, pad_train_x, X_pos_arr, train_y,epochs,batch_size,pad_test_x,X_pos_test_arr,test_y): **def** **word_vector_model(option,sentences,embed_dim,workers,window,train_data,X_train,y_train):** Args:[type={int},Set_Pre-trained_Embedding_options:{0:Self_Embedding, 1:Word2vec,2:Fasttext_2018,3:GloVe,4:Word2vec+PoS,5:Fasttext_2018+PoS, 6:GloVe+PoS} ; sentences = {list} list of tokenized words ;embed_dim = {int} embedding dimension of the word vectors ;workers = {int} no. of worker threads to train the model (faster training with multicore machines); window = {int} max distance between current and predicted word; y_train = y_train] Returns: model = Self_Embedding/Word2vec/Fastext_2018/Glove model trained on the training corpus
5	**Prepare Neural Network Model:** **for each NN option in [0,1,2,3]:** #as an example, we include only MLP (2Layers with Self Embedding Layer) **def classification_model_MLP(embed_dim,X_train_pad,X_test_pad, y_train,y_test,vocab_size,word_index,embed_model, trainable_param,option):** add→ Embedding(vocab_size, embed_dim, weights =[embedding_matrix]) to model add→ Input(shape=(X_train_pad.shape[1],)) add→ embed_layer (input_seq) add→ Dense(256,activation ="relu")(embed_seq) to model add→ Flatten()(x) to model preds = Dense(y_train.shape[1],activation="sigmoid")(x) model = Model(input_seq,preds) loss_param = ["loss"][0], if y_train.shape[1] == 2, loss_param = ["loss"][1], if y_train.shape[1] > 2 return model
6	**Save Model:** combined_model.save(path) #combined model for NN+embedding layer
7	**Test Model and return epochs training history:** predictions = combined_model.predict([x_test_pad, x_test_pos_pad]) return(nn_cell,option,y_test,predicted_class,report_dict,history)

FIGURE 13. Pseudocode of process 5, tweets location prediction (using NN and word embedding).

and the great majority of suggested models are based on character-level CNNs applied to one-hot text vectors or 1D CNNs. These networks perform well when the dictionary size is modest. However, input sequences' one-hot encoding vector size might be rather large for certain languages. To address the aforementioned issues, we use a system in which CNN is configured to analyze the full text in the form of a scanned image. Where we transform our text documents to image scans and train our classifier using the raw pixel values [42] and [79]. This enables us to apply 2-D convolutional layers to text in a rational manner, using advancements in neural network models specifically intended for and targeted at computer vision challenges. This enables us to circumvent the concerns raised before about the usage of 1-D character-level CNNs since document processing now depends on the concurrent extraction of visual characteristics from several lines (depending on the filter size) of text. This technique views text categorization as a problem of learning context-dependent semantic rules with the idea that further semantic information may be recovered from features produced by visual text processing than from strings of abstract discrete symbols. The proposed CNN consists of three convolutional layers, three max-pooling layers, a fully connected layer, and an output layer containing five units/classes. 64-filters have been used in the three convolutional layers of sizes (3, 4, and 5) with stride 2, respectively.

c: RECURRENT NEURAL NETWORKS

Recurrent Neural Networks (RNN) can process words sequentially, as the purpose of RNN training is to predict the next token in a string of words, achieving state-of-the-art performance on a variety of problems, including time series-related problems [42]. The phenomena of bursting and disappearing gradients, which may be generated by increasing the input sequence length, are major problems in normal RNNs. This is the RNN's short-term memory difficulty. To address the short-term memory issue, a new architecture was presented as Long Short Term Memory (LSTM) unit has a more complicated structure that includes a memory cell to retain input dependencies and three gates that operate as regulators: input, output, and in more modern versions, forget gates, which allow the cell to reset its state. The forget gates control how data is delivered into the cell. While this is a significant improvement, input sequence processing is still confined to one direction. A Bidirectional LSTM (Bi-LSTM) [79] may be utilized to get around this issue, as it can process the input sequence both forth and back. However, since both passes must be performed individually, the input sequence cannot be collected concurrently in a bidirectional way. 2 Layers, 256-LSTM, and 256-Bi-LSTM, were implemented based on pre-trained word embedding as explained below.

For all Deep Neural Networks proposed above, MLP and CNN, RNN, the neural network are given word embedding's as input, resulting in the utilization of a 300-dimension vector to represent each word, with longer inputs being truncated and shorter inputs being padded with zero values to ensure that all inputs have the same length for modeling. The experiment uses the top 40,000 most common terms in the dataset to define the vocabulary. This first layer's output would be a 300×40000 matrix. To train deep neural networks using stochastic gradient descent with error backpropagation, an activation function that appears and behaves like a linear function while being a nonlinear function is required. This

allows for the learning of complicated connections in the data. Additionally, the function must be more sensitive to the activation sum input and resist saturation. The output layer consists of a dense layer of five neurons with a Softmax activation function that generates predicted output values. The model was trained using ten epochs and a batch size of 64. Adam algorithm is utilized in training because it provides the optimal answer by regulating the learning rate. Various word embedding was experienced with MLP, CNN, LSTM, and Bi-LSTM networks to investigate which approach produces the highest prediction accuracy, including Word2Vec, Word2Vec with PoS, GloVe, GloVe with PoS, FastText, and FastText with PoS. Additionally, Bi-LSTM networks are utilized with transformer-based word embeddings such as BERT and RoBERTa. This is discussed in depth in the section that follows. Evidently, better accuracy may be attained by training this network with bigger embedding dimensions and adding additional hidden layers, but this increases the necessary training time.

d: NEURAL NETWORK-BASED (STATIC) WORD EMBEDDING

Word embedding is an n-dimensional vector space representation of words that uses training data to determine which words are semantically similar or related in vector space. Individual words are represented in a vector space as real-valued vectors. Each word is assigned to a vector, and the vector values are learned using a similar process to that of a neural network. This process employs densely distributed word representations. Each word is a tens-or-hundreds-dimensional vector. In comparison, sparse word representations like one-hot encoding require thousands or millions of dimensions. This lets words with similar meanings have comparable representations.

This contrasts with the crisp but fragile representation in a bag of words approach, where words have multiple representations unless explicitly specified. Word2vec, GloVe, and FastText are unsupervised learning-based word embedding algorithms. Word2vec is a two-layer simple neural network that takes as input a corpus of texts and produces a vector for each word in the corpus. Words that frequently occur in similar contextual locations throughout the corpus and also occur frequently in similar word2vec spaces [89]. Word2vec effectively preserves semantic linkages, as words with similar neighbors are likely to be semantically related. There are two types of word2vec embedding. The continuous bag of words (CBOW) technique includes n-words preceding the target word and m words after it. The Skip-Gram model forecasts adjacent words based on the current word [103]. The gloVe is another well-known unsupervised word embedding technique that is also based on the distributional hypothesis: ''words that appear in comparable settings are likely to have similar meanings.'' The gloVe is distinct from Word2vec in that it generates vectors of words based on their co-occurrence data [90]. Word2Vec is more predictive than GloVe; for example, in the Skip-gram setting, it attempts to ''guess'' the correct target word based on its context words. GloVe, on the other hand, is a count-based algorithm. It starts by generating a matrix X with rows representing words and columns representing contexts, and the element value Xij equals the number of times a word i appears in a context, including a word j. By minimizing reconstruction loss, this matrix is factorized into a lower-dimensional representation, with each row representing the vector of a specific word. Thus, Word2Vec learns word vector representations based on their local context, whereas GloVe learns word vector representations based on global statistics on word co-occurrence. A group of researchers working at Facebook have introduced FastText to improve the Word2Vec paradigm [91].

In contrast to the Word2Vec framework, which is trained to generate vectors for individual words, the FastText framework is trained to generate numerical representations of character n-grams. Thus, words are a collection of character n-grams. For example, if we consider the word ''virus'' and utilize n = 3 or tri-grams, we get the following n-grams: ''vi'', ''vir'', ''iru'', ''us >'', as ''<'' and ''>'' are special characters that appear at the beginning and end of each word.

FastText's character embedding enables it to construct embedding vectors for words that are not even included in the training texts. However, FastText has one significant drawback: the high memory needs to be associated with the process of building word embedding vectors from its letters. Meanwhile, it features high-speed processing, which minimizes the total required training time. For Part-of-Speech (PoS) consideration in our implementation with the pre-trained word embedding, we have generated a one-hot vector for the PoS tag. Then, for each word embedding, a representation consisting of its word embedding is concatenated with its PoS tag. Most of the development in the NLP domain may be credited to general deep learning research improvements. In recent years, word embeddings that consider context have received great attention for various natural language processing tasks. In particular, Google has developed a unique neural network design known as a transformer, which offers numerous advantages over typical sequential models (LSTM, RNN, GRU, etc.) [104]. The benefits include, but were not limited to, more effective modeling of long-term relationships among tokens in a temporal sequence and more efficient model training in general by reducing the sequential dependence on prior tokens. A transformer, in a nutshell, is an encoder-decoder architectural model that employs attention methods to convey a full representation of the whole sequence to the decoder at once rather than sequentially [105]. In this research, we considered implementing two transformer models, BERT and RoBERTa. Both will be explained in detail in an upcoming section.

e: TRANSFORMERS-BASED (DYNAMIC) WORD EMBEDDING

A transformer is a deep learning model that employs a distinct method of weighing the significance of each component of the input data [105]. On the other hand, recurrent neural networks (RNNs) are designed to handle sequential input data, such as natural language, for tasks such as translation and

text summarization. Transformers, do not require to process data in the same order as RNNs. As a result, the attention method can be used to contextualize any point (index) in the input sequence. Such that, the transformer does not pursue to process the beginning of the input sequence before its end; instead, it detects the context in which each element in the sequence receives its meaning [105]. This feature allows for more parallel computation than RNNs, reducing training time. Transformers have now become the preferred approach for natural language processing tasks since their introduction in 2017, replacing RNN models such as long short-term memory (LSTM). Because of the adoption of training parallelization, training on larger datasets is now possible. As a result, pre-trained methods such as BERT (Bidirectional Encoder Representations from Transformers) and RoBERTA (Robustly Optimized BERT Pretraining Approach) were developed, which were trained and fitted using massive language datasets and may be fine-tuned for specific applications. Our experiments employ transformer models, which may be found in Hugging Face's transformer library. For the classification job, the embedding would be input into a recurrent neural network that has two layers of bidirectional long-short-term memory (Bi-LSTM-256). Finally, the final hidden state from the Bi-LSTM layer would be sent into a max-pooling layer to remove unnecessary information. Subsequently, the feature maps will be concatenated and input into a linear fully connected layer, which will then output the classification result. The recurrent neural network has a dropout rate of 0.1. Due to the fact that we employ fixed-size input to the models, we either add padding at the end or eliminate unnecessary tokens if the total number of tokens exceeds the specified sequence length. Tokens for padding are disguised throughout training to ensure their absence. These tokens are used as input for the transformer models, which pass them via many layers to create the final hidden embedding for each input token. While performing the classification job, we examine just the first token corresponding to the sequence token's beginning.

BERT

BERT is a bi-directional transformer used for pre-training over large amounts of unlabeled text data in order to learn a language representation that may be fine-tuned for specific machine learning applications and tasks [92]. While BERT surpassed the NLP state-of-the-art on multiple tough tasks, the bidirectional transformer, unique pre-training tasks of Masked Language Model and Next Structure Prediction, a large amount of data, and Google's compute capacity all contributed to its improved performance.

Language models typically scan input sequences from left to right or right to left. This type of one-way training works well for predicting/generating the next word. However, BERT employs Transformer's bidirectional training to gain a deeper awareness of language context. It's also called ''non-directional''.

So, it considers both previous and subsequent tokens. Generally, Transformer Network includes both encoder and decoder; the encoder reads and processes text, while the decoder is in charge of predicting the task. In their paper [92], Google's researchers introduced two models, BERT-base and BERT-large, such that BERT was trained using 3.3 billion words from Wikipedia and BooksCorpus. It employs Masked Language Modelling (MLM) and NSP (Next Sentence Prediction). Many words in the MLM sequence are randomly masked with the token [MASK]. So, it can guess masked words from the remaining words' context. As for NSP, BERT employs NSP training to grasp sentence relationships. The model is fed pairs of sentences and trained to predict whether the second sentence follows the first or not.

RoBERTa

Introduced by Facebook, RoBERTa [93], which stands for Robustly Optimized BERT Pretraining Approach, is one of the most innovative architectures to originate from the BERT revolution. While BERT provided a substantial performance improvement across several tasks, RoBERTa proposed retraining of BERT with an improved training methodology. The authors in [93] proposed several improvements to the original BERT design in order to achieve better performance. Among other things, RoBERTa eliminates the Next Sentence Prediction (NSP) task from BERT's pre-training and incorporates dynamic masking, which causes the masked token to alter during the training epochs. It was also discovered that larger batch-training sizes were more beneficial in the training operation. It is important to note that RoBERTa utilizes 160 GB of text for its first training. Additionally, RoBERTa uses a different tokenizer than BERT, Byte-Pair Encoding (BPE) (same as GPT-2), and has a bigger vocabulary (50k vs 30k). The authors acknowledge that having a larger vocabulary that enables the model to represent any word results in a greater number of parameters (+15 million for base RoBERTA). However, the authors argue that performance advantages justify the increased complexity.

3) TIME SERIES DATA SYNCHRONY ESTIMATION METHODS

It is a common methodology to use cross-correlation in two-time series to examine the temporal association between event observations in two different contexts. A vector of sequential events is selected from each time series so the number of events in each vector is equal. The Pearson product-moment correlation for these two vectors is then computed. One-time series may be lagging or leading, depending on the start time of the vectors. Lag or offset is the time between two vectors' events start [106].

A window is a vector of sequential observations sampled from a time series. In order to quantify the cross-correlation of two-time series, each of which contains (O) observations, with equal intervals of time between observations (p). Cross-correlation between M and N at a lag (d) can be expressed as a function $C\ (M, N)$, which can be defined in its simplest form if we assume both time-series stationarity and choose

to consider positive lags of observations [106] as presented in (1).

$$C\,(M,N) = \frac{1}{O-d}\sum_{i=1}^{O-d}\frac{(M_i-\overline{M}).(N_{i+d}-\overline{N})}{StdDev\,(M)\,.StdDev\,(N)} \quad (1)$$

In which,

$\overline{M}, \overline{N}$ are: the time series means, and

$StdDev\,(M)\,, StdDev(N)$ are the standard deviations of time series M and N, respectively.

This can be described as a simple Pearson correlation between the two-time series, with their events observed to be lagged by d time instances. Outliers can skew the findings of the correlation estimation, also, Pearson correlation implies that the data are homoscedastic, meaning that the variance of the data is uniform across the data range. Generally, Pearson correlation is often a brief assessment of global synchrony. Therefore, it provides no directionality information between the two-time series, such as which time series precedes the other. Nevertheless, the Pearson correlation provides a method that may be used to examine the locally available moment-to-moment synchrony. This can be computed in several ways, one of which involves measuring the Pearson correlation in a limited-size window of the time series and then continuing the procedure along with a rolling window until the entire time series has been analyzed. This requires an arbitrary definition of the window size at which you would like the method to be repeated, which can be negatively subjective [107]. A leader-follower relationship, in which the leader initiates an event that a relevant event by follower follows, is an example of a situation in which time-lagged cross-correlation (TLCC) can be used to determine the directionality of the link between two-time series. Moreover, It is possible to compute the windowed time-lagged cross-correlations in order to evaluate the dynamics on a more granular scale (WTLCC). During this process, the time-lagged cross-correlation is repeated in several different time series windows. Then, we can analyze each window, or we can use the total of all of the windows to arrive at a score that compares and contrasts the differences in the leader-follower relationship between two different series [107]. In our proposed framework, we have depicted the fine-grained dynamic interaction between two-time series, such as the leader-follower association and how it shifts over time, by employing time-lagged cross-correlations and overlapping-windowed time-lagged cross-correlations. The window size, overlap window, and window position along the two-time series were decided concerning the emergence and development of COVID-19 outbreak in the most hazardous geolocations as per the number of infected cases and tweets volumes and sentiments as extracted from the framework'sprecedingg stages. Such that, we have decided to study the synchrony between the time series representing the development of daily active cases of infection and the volumes of negative sentiment tweets volumes in Australia, Canada, India, the UK, and Uthe SA, taking into consideration the time series window that is most relevant to the peak of active cases counts in two different disease progression waves as shown in the Table 5.

TABLE 5. Corona virus pandemic (Covid-19) statistics of selected countries till dec21 [108].

Country	2020 Main Peak	2021 Main Peak	Total Number of Infections	Pop.
Australia	July – Sept.	July – Sept.	684K	26 M
Canada	Nov, Dec, Jan21	Apr – June	2.55M	38 M
India	July – Sept.	Apr – June	35.5M	1380M
UK	Nov, Dec, Jan21	July – Sept.	14.3M	68 M
US	Nov, Dec, Jan21	July – Sept.	60M	330 M

Assume we have two time series in the form of a one-dimensional vector, each with O observations and equal time spans of length p between observations. Assume a window of w_{max} span, a lag period of d on the span $d_{max} \leq d \leq +d_{max}$, and an elapsed interval index i from the onset of the data vector. For each i a couple of windows W_M and W_N can be chosen from the two data vectors M and Y, [106], such as presented in (2) and (3):

$$W_M = \begin{cases} \{M_i, M_{i+1}, \cdots\cdots M_{i+w_{max}}\}, & d \leq 0 \\ \{M_{i-d}, M_{i-d+1}, \cdots\cdots M_{i-d+w_{max}}\}, & d > 0 \end{cases} \quad (2)$$

and,

$$W_N = \begin{cases} \{N_{i+d}, N_{i+d+1}, \cdots\cdots N_{i+d+w_{max}}\}, & d \leq 0 \\ \{N_i, N_{i+1}, \cdots\cdots N_{i+w_{max}}\}, & d > 0 \end{cases} \quad (3)$$

Cross-correlation between W_M and W_N can be expressed as a function $C(W_M\ ,W_N)$, which can be defined as presented in (4):

$$C\,(W_M, W_N) = \frac{1}{w_{max}}\sum_{i=1} w_{max}\frac{(W_{M_i}-\overline{W_M}).(W_{N_i}-\overline{W_N})}{StdDev\,(W_M)\,.StdDev\,(W_N)} \quad (4)$$

In which,

$\overline{W_M}, \overline{W_N}$ are the window observation means, and

$StdDev\,(W_M)\,, StdDev(W_N)$ are the standard deviations of time series W_M, W_N, respectively.

We guarantee a mirror symmetry by picking the windows in accordance with Equations 2 and 3, which ensures that the resultant cross-correlations coefficients set, expressed as it ranges from -d_{max} to + d_{max}, would contain matching values in the opposite direction even when the instances in W_M and W_N are exchanged. Hereafter, the Pseudocode of Process 6, Quantifying Synchrony (Cross-correlation) of two-time series is presented as shown in Fig. 14.

Process 6: Quantifying Synchronry of two-time series

1 **Load Dataset:** df = pd.read_csv('synchrony_dataset.csv')

2 **Using Windowed time lagged cross correlation:**
Env Param: months = 18, avg_days_per_month = 30, no_of_splits = 9
no_of_samples_per_split = df.shape[0]/no_splits
ccc=[] #cross-correlation coefficients
for t in range(0, no_of_splits):
data1 = df['COVID_Daily_Active_Cases'].
data2 = df['Daily_Negative_Sent_Count'].
evaluate_cc=[crosscorr(data1,data2, lag) for lag in_range of (-int(months*avg_days_per_month),int(months*avg_days_per_month +1))]
ccc.append(evaluate_cc)

3 **Using Overlapping window time lagged cross correlation**
#one combination of the parameters (numerous combinations were examined)
Env Param: months = 18, avg_days_per_month=30,window_size = 60, window_start = 0, window_end = window_start + window_size, step_size = 6
ccc=[] #cross-correlation coefficients
while t_end < 480:
data1 = df['COVID_Daily_Active_Cases'].iloc[window_start:window_end]
data2 = df['Daily_Negative_Sent_Count'].iloc[window_start:window_end]
evaluate_cc = [crosscorr(data1,data2, lag, wrap=False) for lag in_range of (-int(months*avg_days_per_month),int(months*avg_days_per_month +1))]
ccc.append(evaluate_cc)
window_start = window_start + step_size, window_end = window_end + step_size

FIGURE 14. Pseudocode of process 6, quantifying synchrony (Cross correlation) of two-time series.

F. EVALUATION METRICS

Average Weighted Accuracy, Precision, Recall, F1-score, and Confusion Matrix were used to assess the classification performance since these metrics could better reflect the performance of unbalanced classes. Accuracy, Precision, Recall, F1-score formulas, and the way of calculating the weighted metrics are presented in (5)-(9), respectively.

Accuracy: is often used as the base evaluation metric for classification models.

Precision: is a measure of how many positive predictions are correct.

Recall (Sensitivity): is a measure of how many of the actual positive cases the classifier correctly predicted.

F1-score: is the harmonic mean of both precision and recall. Higher F1 score denotes higher testing accuracy.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = 2 \times \frac{Precision + Recall}{Precision \times Recall} \tag{8}$$

$$Weighted\_Metric = \frac{(M_{c1} \times |c1|) + (M_{c2} \times |c2|)}{|c1| + |c2|} \tag{9}$$

In which, $TP, TN, FP, FN, M_{c1}, c1, M_{c2}, and\ c2$ are True Positive, True Negative, False Positive, False Negative instances, evaluation metric for first-class, count of first-class instances, evaluation metric for second class, and count of second class instances, respectively.

These metrics were employed to evaluate the performance of the first and second subsystems in the proposed framework. Additionally, a confusion matrix is used to illustrate classifier performance based on the aforementioned four values (TP, FP, TN, FN). The weighted F1 considers the metrics for each class and averages them by the count of true instances for each label). This modifies the F1 score to account for class imbalance. For the third subsystem, we have measured the cross-correlation coefficient. When investigating time, cross-correlation is the degree of Synchrony between two-time series, or how well one series predicts the other, whereas temporal lag or lead can be considered.

IV. RESULTS AND DISCUSSION

As previously stated, the purpose of this study is to assess social media data's potential to function as an outbreak prediction and identification tool in the context of the emerging COVID-19 pandemic, particularly at the phase of its initial evolution prior to government intervention. We wanted to find out if the total tweets' volume associated with one particular Twitter API query would be employed as a surrogate measure to model the progression of COVID-19 active case counts during the early stages of an increasingly globalized catastrophe. Also, the capacity to gauge public opinion and link trends with reported cases during the first epidemic is studied by analyzing the linguistic features of tweet content to see whether there is any correlation. With the help of data scraped from the Twitter social network related to the COVID-19 pandemic, we were able to determine the most affected geolocations, which were then used as inputs to a variety of classic and up-to-date classification techniques to aid in the reduction of the virus's spread through the implementation of a proactive pandemic awareness system. The source data for all of the analysis conducted in this study contains open-source, publicly available tweet data that was precisely scraped for the purposes of this research. Using sentiment analysis, tweets were first categorized, and then the development of positive, negative, and neutral sentiment progression was investigated. This exploratory analysis revealed a substantial daily increase in the count of tweets (of all sentiments) in the time periods preceding the emergence of the new COVID-19 infectious waves, which ran from March 2020 to September 2021. In this study, we offer a textual analysis of Twitter data in order to determine the public community sentiment, which has been linked to the fast progression of COVID-19. In particular, the research makes a significant addition by conducting a comparison analysis for the prediction of tweet geolocation using various machine and deep learning models. Consequently, we conducted several tests to verify our proposed model and extract location information from COVID-19-related tweets to acquire reliable findings.

We employed 10-fold cross-validation to reduce the possibility of bias. The average of the findings from each of the folds is used to assess the overall system performance. The results are summarized such that a) an exploratory data

analysis is presented and, a sentiment prediction of all tweets is computed in the next subsection, b) The results for geolocation prediction using classical machine learning methods and deep learning methods are depicted in the following subsection. These were obtained by employing the methods and processes implemented in our proposed first and second subsystems, using different feature vectors per each sub-problem's requirements. Finally, the crowd situational awareness computations and visualizations are proposed in the last subsection.

A. EXPLORATORY DATA ANALYSIS AND CHRONOLOGICAL SENTIMENT PREDICTION

This section explores many insights extracted from the cleaned and analyzed tweets. As it was declared earlier, we have collected around 577,000 unique tweets distributed amongst 124 countries. Fig. 15 shows the distribution of collected tweet volumes and the histogram of the top 10 countries with the highest count of collected tweets, indicating that the United States is the highest, followed by the United Kingdom, Canada, India, and Australia. These five countries were considered the most likely to be affected by the COVID-19 outbreak, and their data will be used for further examinations. It was observed that late-year 2019 and year 2020 witnessed an obvious increase in the number of created Twitter accounts compared to previous years' counts, as shown in Fig. 16.

Additionally, Fig. 17 and Fig. 18 show the tweet length distribution and the top 10 hashtags as mentioned in the scraped dataset, respectively. Our study found that the percentage of tweets with negative, neutral, or positive sentiment about a given issue fluctuated over time when investigated using a hybrid sentiment analysis model, relying on both rule-based and ML-based methods. The rule-based layer relies on the Valence Aware Dictionary for Sentiment Reasoning (VADER) paradigm. This text sentiment analysis model is sensitive to both polarity (positive or negative) and the degree of emotion. VADER computes a normalized score (or "compound score") by adding the valence score of each word in the input text. Followed by an ML-based layer, using many conventional machine learning classifiers alternatives, such as Random Forest Classifier, Ada-Boost Classifier, MultinomialNB, Decision Tree Classifier, Extra Trees Classifier, and Gradient Boosting Classifier, to assess the sentiment in tweets.

These classifiers' performance was tested in relation to the accuracy of the sentiment prediction compared to the VADER sentiment output. Different feature selection methods have been tested, such as count vectorizer, n-grams, and TF-IDF, as presented in Table 6. For both the count vectorizer and the TF-IDF, the Decision Tree Classifier outperformed other models on the validation dataset with an accuracy of 92.4% and 94.3%, respectively. For the bi-gram feature extraction model, Extra Trees Classifier outperformed other models with an accuracy of 88.3%. The proposed framework outperformed comparable research as per our literature, as indicated in Table 7.

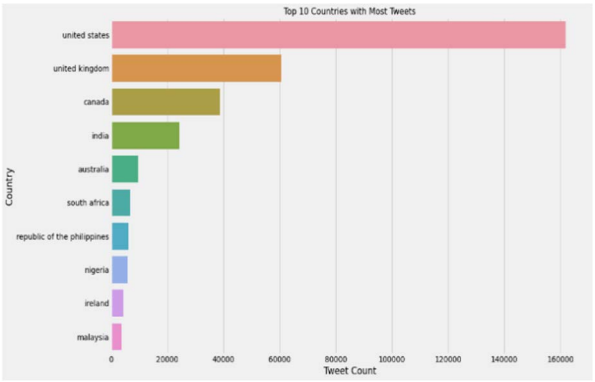

FIGURE 15. Top 10 countries according to the count of tweets in the scraped dataset.

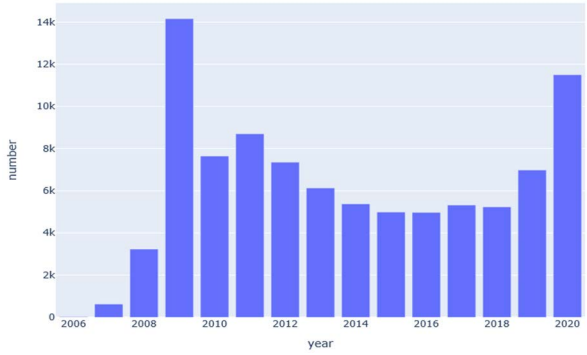

FIGURE 16. Count of twitter users created since 2006 to 2020.

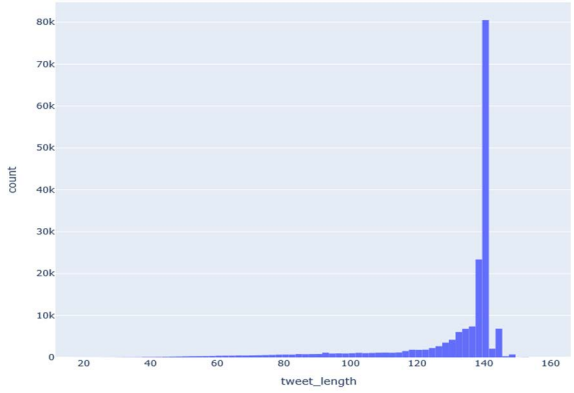

FIGURE 17. Tweets length distribution according to the scraped dataset.

Hence, in our implementation, ML-Based sentiment classification was built on top of a lexicon-based sentiment classifier which scores each word in the input text; we have observed that using unigram features has resulted in higher precision than n-gram. Furthermore, TF-IDF features vectorization outperformed count vectorization because it gives more weight to features that appear in a single tweet or document rather than commonly repeated features across the

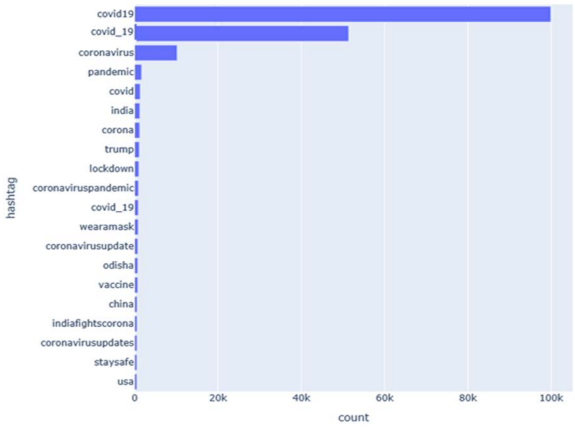

FIGURE 18. Top 20 hashtags according to the scraped dataset.

TABLE 6. Machine based sentiment classifiers, performance comparison.

Model	Unigram+ Count Vectorizer	Ngram+ Count Vectorizer	Unigram +TF-IDF
RandomForestClassifier	80.3%	77.0%	83.4%
AdaBoostClassifier	81.6%	77.8%	83.6%
MultinomialNB	78.4%	77.4%	79.9%
DecisionTreeClassifier	92.4%	87.8%	**94.3%**
ExtraTreesClassifier	89.9%	88.3%	90.8%
GradientBoostingClassifier	45.2%	43.2%	43.1%

entire set of tweets or documents. This appears to have aided in improving feature representation and resolving remaining data preprocessing issues. The sentiment analysis of the collected tweets revealed that 44% of the tweeting users are positive about the COVID-19 outbreak, 40% are neutral, and 16% are negative, according to the DecisionTreeClassifier+ TF-IDF results.

B. CHRONOLOGICAL SPATIAL ANALYSIS AND PREDICTION

Despite the importance of geo-location data in social media event detection, only a few social media datasets include information about users' location. Geographic coordinates (i.e.., longitude and latitude) can be used to pinpoint exact locations in social media posts, although the information can also be described in a more general way, such as ''anywhere around here'' (i.e., city name). So, in addition to our textual analysis of Twitter data to ascertain public sentiment, which has been associated with the rapid spread of COVID-19 infections in recent months, the study contributes significantly by doing a comparative analysis of tweet location recognition using a number of machine and deep learning models. Accordingly, we ran a series of experiments to validate our proposed model and extract location information from COVID-19-related tweets to get trustworthy results.

TABLE 7. Proposed framework vs. literature results comparison.

Research Work	Accuracy	EDA
Proposed Framework, Sub-system 1: Sentiment Analysis and Feature Selection	94.3%	Worldwide: April-May 2020 44% Positive, 40% Neutral, 16% Negative
Kamaran Mangur, 2020	-	Worldwide: April 2020 36% Positive, 50% Neutral, 14% Negative
Bishwo Prakash Pokharel, 2020	-	Nepal: May 2020 58% Positive, 27% Neutral, 15% Negative
Muvazima Mansoor, 2020	84.5%	-
Ali Shariq Imran, 2020	82%	-
CHIUNG CHING HO, 2020	87.7%	-

TABLE 8. Classical machine learning-based geolocation classifiers, models' performance comparison.

Model	Unigram+ Count Vectorizer	Ngram+ Count Vectorizer	Unigram +TF-IDF
RandomForestClassifier	66.8%	66.7%	71.6%
AdaBoostClassifier	65.9%	66.8%	68.4%
MultinomialNB	72.7%	73.2%	75.9%
DecisionTreeClassifier	71.2%	70.4%	74.2%
ExtraTreesClassifier	76.8%	76.6%	**80.8%**
GradientBoostingClassifier	30.4%	27.6%	30.1%

1) CLASSICAL MACHINE LEARNING ALGORITHMS

Following the sentiment classification using hybrid methods, we have implemented a location prediction approach using the same textual data scraped from Twitter with the help of conventional machine learning-based classifiers such as Random Forest Classifier, Ada-Boost Classifier, MultinomialNB, Decision Tree Classifier, Extra Trees Classifier, and Gradient Boosting Classifier. Count vectorizer, n-grams, and TF-IDF are some of the feature selection approaches that have been evaluated. ExtraTreesClassifier outperformed other models on the validation dataset for all feature extraction approaches with an accuracy ranging from 76.6% to 80.8%. Even though the same datasets were used, location prediction or classification is a totally distinct challenge from sentiment classification. In order to get adequate performance for this problem, certain features and methods must be implemented. As a result, we were unable to achieve a precision of more than 80% using typical machine learning algorithms, despite applying a variety of feature selection and vectorization methods, as shown in the accompanying Table 8.

Consequently, more complex algorithms and approaches such as MLP, RNN, and CNN, empowered with pre-trained word embedding, were investigated in order to produce

considerably more accurate and trustworthy location predictions presented in the upcoming section.

2) DEEP LEARNING ALGORITHMS

In addition to the classical machine learning approaches, we have proposed many deep learning models such as Simple Multi-Layer Perceptron Neural Network (MLP), Convolutional Neural Network (CNN), Long Short-term Memory (LSTM), and Bidirectional Long Short-term Memory (Bi-LSTM) to tackle the problem of predicting tweets' geolocation by extracting features within the tweets such as tweet text and tweet sentiment and the features associated with the tweets such as the user name, and user description, user location. The task of predicting a tweet's location can be approached as a classification problem, with the goal of predicting country labels for a single tweet. In implementing and evaluating the classification models, we only included tweets relevant to the five main countries under investigation. Many word embedding algorithms, including Word2Vec, FastText2018, and GloVe, were used to generate feature vectors in the form of word embedding, both individually and in conjunction with PoS tagging. Two additional tests were carried out by implementing a transformer-based word embedding using BERT and RoBERTa. In order to identify semantic similarity and conduct classification tasks, this text embedding transforms tweet texts into high-dimensional vectors. After obtaining the word embedding for each tweet, a method was used to balance the five classes. To achieve this balance, we selected the class with the least amount of samples, then reduced the size of the remaining classes to a comparable size (downsampling). This ascertained that the training was not skewed toward a particular class. The data were randomly shuffled and divided into training and validation sets, with 90% serving as the training dataset. Multiple classifiers were used to achieve the best performance, which was evaluated using accuracy metrics.

Considering that all presented experiments and implementation test runtime are compiled and evaluated using Google Colab Pro-environment with up to 25 Gbytes RAM and K80, P100, and T4 GPUs. Performance metrics in the form of accuracy, precision, recall, F1 score, and computation time per 10 epochs are presented in the following Table. 4 to compare the implemented combinations of deep neural network models with different word embedding models used for location classification. Some of our results and graphical illustrations are presented below, including a comparison between training and validation accuracy, a comparison between training and validation loss, and the confusion matrix associating the actual and predicted geolocations.

From Table 9, Figs. 19 to 21, it can be realized that, the Multi-Layer Perceptron (MLP) deep neural network with self-embedding layer, word2vec, and Glove pre-trained word embeddings resulted in very poor accuracy of 10%, 36%, and 32%, respectively, while adding the PoS tag one-hot encoding embedding assisted in elevating the validation accuracy of the same model using word2vec and Glove to about 89%.

TABLE 9. Deep learning-based geolocation classifiers, models' performance comparison.

DNN Model	Word Embedding	Acc.	Prec.	Recall	F1	Computation Time (Minutes) /10epochs
MLP	Embedding Layer	0.1	0.11	0.12	0.10	412
	Word2Vec	0.36	0.13	0.36	0.19	17
	Word2Vec +PoS	0.89	0.90	0.89	0.90	17.5
	FastText	0.89	0.90	0.89	0.89	16.7
	FastText+ PoS	0.90	0.91	0.90	0.90	17.4
	GLOVE	0.32	0.11	0.32	0.18	18
	GLOVE+ PoS	0.89	0.89	0.89	0.89	18.3
CNN	Embedding Layer	0.92	0.92	0.92	0.92	62
	Word2Vec	0.91	0.91	0.91	0.91	23
	Word2Vec +PoS	0.90	0.90	0.90	0.90	17.5
	FastText	0.91	0.91	0.91	0.91	21
	FastText+ PoS	0.90	0.90	0.90	0.90	17
	GLOVE	0.91	0.91	0.91	0.91	21
	GLOVE+ PoS	0.90	0.90	0.90	0.90	18
LSTM	Embedding Layer	0.90	0.89	0.90	0.89	374
	Word2Vec	0.36	0.13	0.36	0.19	111
	Word2Vec +PoS	0.90	0.91	0.90	0.90	113
	FastText	0.34	0.12	0.34	0.17	114
	FastText+ PoS	0.91	0.91	0.91	0.91	110
	GLOVE	0.32	0.12	0.32	0.17	115
	GLOVE+ PoS	0.90	0.91	0.90	0.90	118
Bi-LSTM	Embedding Layer	0.91	0.91	0.91	0.91	109
	Word2Vec	0.36	0.13	0.36	0.19	30
	Word2Vec +PoS	0.87	0.88	0.87	0.87	30
	FastText	0.90	0.90	0.90	0.90	31
	FastText+ PoS	0.90	0.90	0.89	0.89	28
	GLOVE	0.32	0.12	0.32	0.17	33
	GLOVE+ PoS	0.88	0.87	0.88	0.87	34
Bi-LSTM	BERT	0.93	0.92	0.93	0.93	240
Bi-LSTM	RoBERTa	0.96	0.96	0.96	0.96	275

Meanwhile, MLP with a self-embedding layer resulted in the worst accuracy and the highest computation time and resource requirements among all experiments. Using FastText and FastText+PoS tagging with MLP resulted in

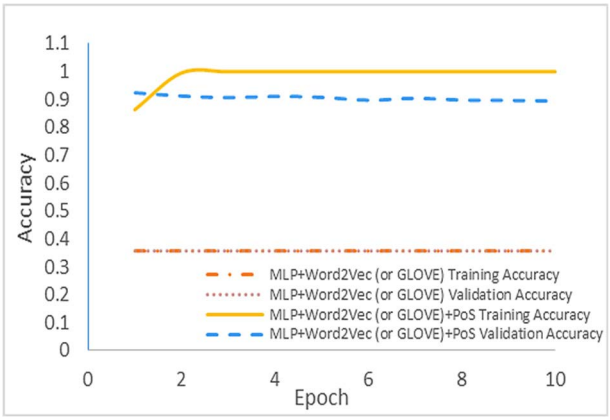

FIGURE 19. MLP with Word2Vec (or GLOVE) classifier accuracy.

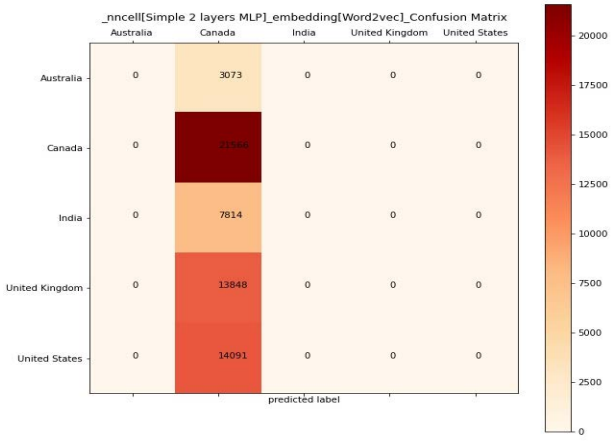

FIGURE 20. MLP with Word2Vec (or GLOVE) classifier conf. matrix.

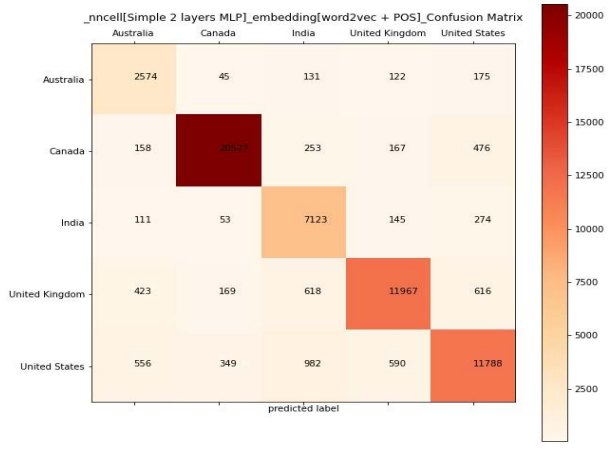

FIGURE 21. MLP with Word2Vec+PoS (or GLOVE+PoS) classifier conf. matrix.

an accuracy of 89% and 90%, respectively, as shown in Fig. 22 and Fig. 23.

From Table 9, Figs. 24 to 27, it can be comprehended that Convolutional Neural Network (CNN) with self-embedding layer, word2vec, Glove, and FastText pre-trained word

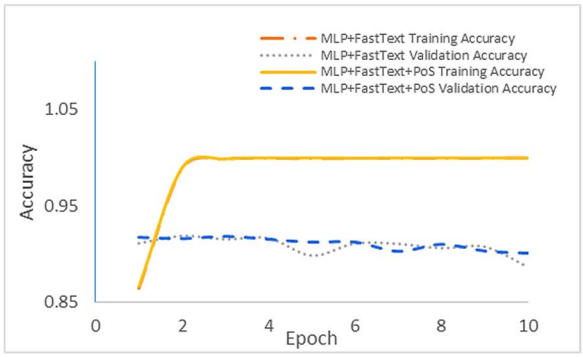

FIGURE 22. MLP with FastText and FastText+PoS classifier accuracy.

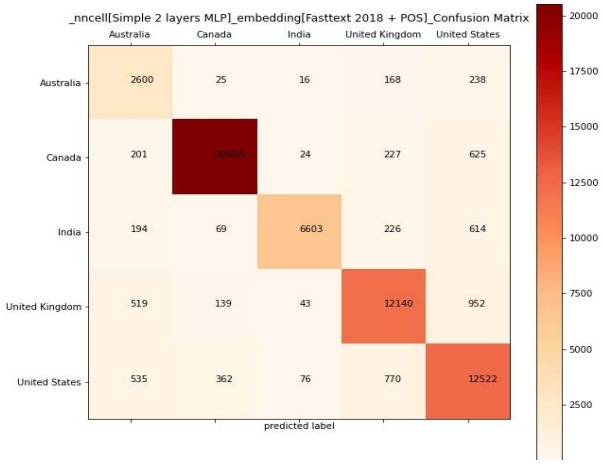

FIGURE 23. MLP with FastText+PoS classifier conf. matrix.

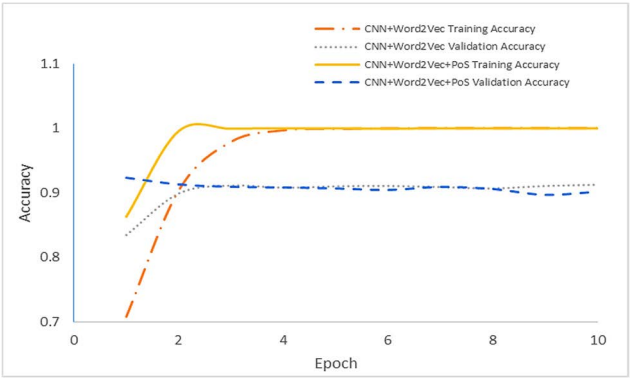

FIGURE 24. CNN with Word2Vec and Word2Vec+PoS classifier accuracy.

embeddings resulted in an accuracy of 90% to 92%. Using the self-embedding layer with the CNN implementation resulted in the highest accuracy among the previously mentioned embeddings, but it required three times the computation time and resources of the pre-trained embeddings. Furthermore, it is recognized that employing PoS in conjunction with word embeddings helped in reducing the computation requirements of CNN.

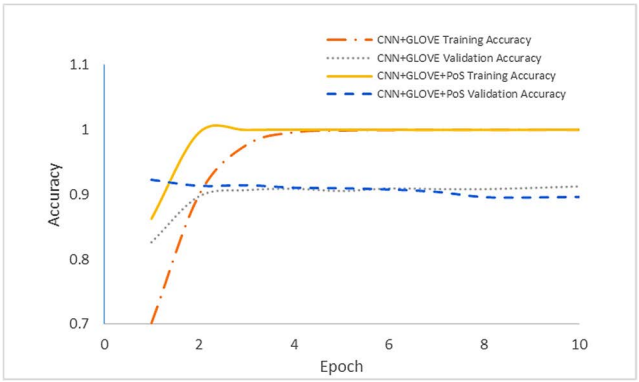

FIGURE 25. CNN with glove and Glove+PoS classifier accuracy.

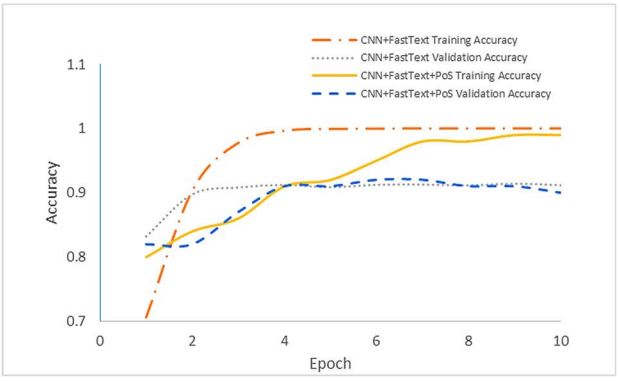

FIGURE 26. CNN with FastText and FastText+PoS classifier accuracy.

The same results can be recognized for LSTM and Bi-LSTM with self-embedding layer and pre-trained embeddings, as presented in Table 9. Nevertheless, as previously stated, despite their satisfactory performance, some of our proposed algorithms have high computational complexity in terms of required processing resources and training time, such as LSTM with embedding layer, and Bi-LSTM with RoBERTa transformer-based models, which have the highest average computation time per step of 792 ms/step, 724 ms/step, and 522 ms/step, respectively (as per our implementation parameters and data size, each epoch =∼3100 step) as displayed in the following Fig. 28 and Fig. 29. Evidently, employing pre-trained word embedding significantly reduced the total computation time necessary to achieve elevated accuracy when compared to that attained by implementing DNN models with the assistance of a self-embedding layer. This method performs particularly well on our domain-dependent data when Parts of Speech is used to identify the Nouns and Adverbs in the data. The incorporation of additional aspect terms in addition to the basic data vectorization improved the performance of the pre-trained models. Amongst all experiments, models with FastText+PoS word embedding outperformed or were equivalent to other pre-trained word embedding's when measured in terms of accuracy as it scored 90-91% in all cases as presented in Table. 4.

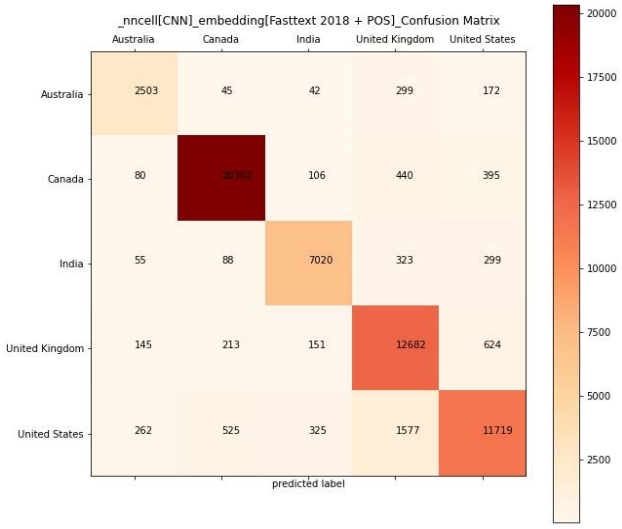

FIGURE 27. CNN with FastText+PoS classifier conf. matrix.

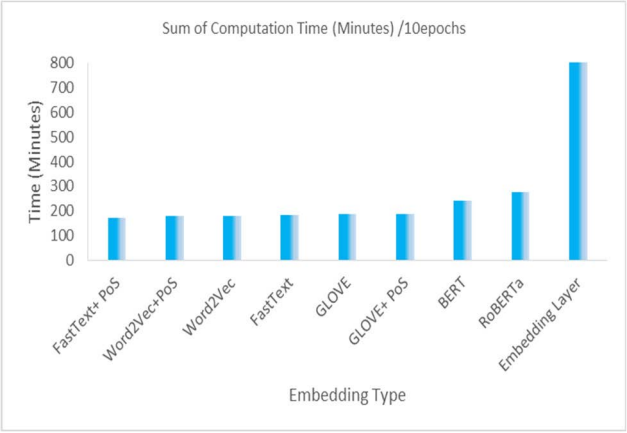

FIGURE 28. Sum of computation time (Minutes) /10epochs, by word embedding algorithm.

Also, the higher accuracy obtained by models with FastText and FastText+PoS could be referenced to the fact that it supports n-gram vectorization rather than word vectorization. This can be explained that the vectorization approach used in FasText was able to recognize and understand the various ways of writing location-specific terms provided by users with varying educational, cultural, and social backgrounds. At the same time, it outperformed all other models when evaluated in terms of necessary computing resources (processing and time requirements). Such that models with pre-trained word embedding's needed less than one-third of the needed computing time elapsed using the self-embedding layer. Furthermore, models using FastText+PoS were able to minimize the validation loss faster than other proposed experiments, allowing it to be employed for fewer epochs to achieve the same accuracy while saving significant amounts of computation resources, as presented in Fig. 30, Fig. 31, Fig. 32 and Fig. 33. Likewise, Bi-LSTM with RoBERTa embedding resulted in rapid minimization of validation loss

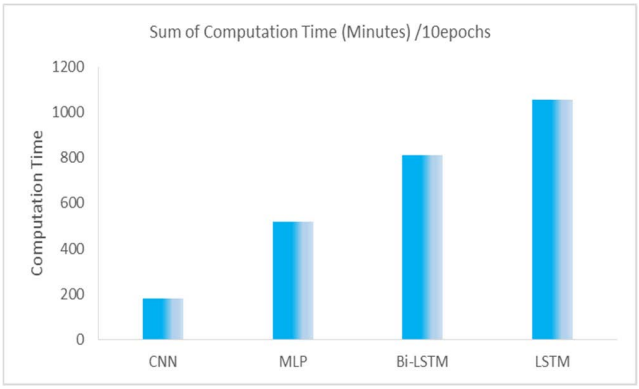

FIGURE 29. Sum of computation time (Minutes) /10epochs, by deep neural network model.

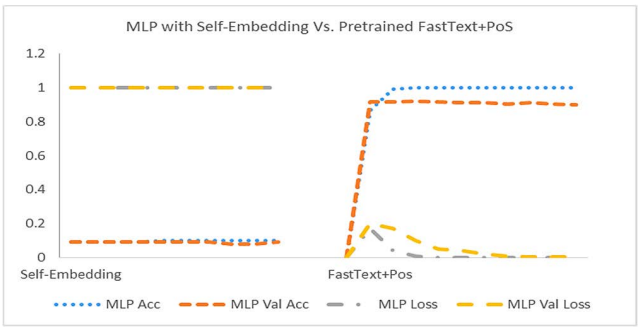

FIGURE 30. MLP with self-embedding vs. MLP with FastText+PoS training and validation accuracy and loss changes during 10 epochs.

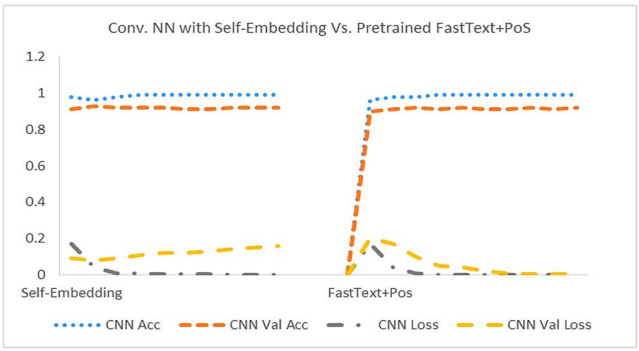

FIGURE 31. CNN with Self-Embedding vs. CNN with FastText+PoS training and validation accuracy and loss changes during 10 epochs.

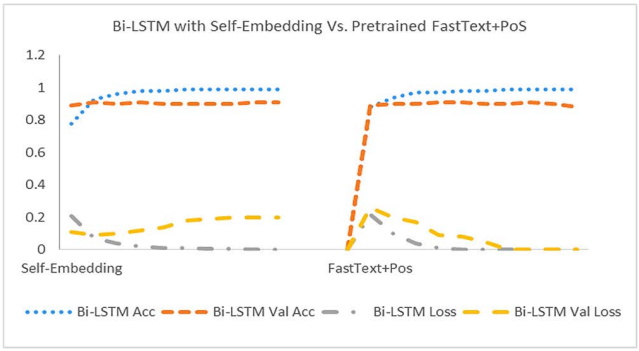

FIGURE 32. Bi-LSTM with Self-Embedding vs. Bi-LSTM with FastText+PoS training and validation accuracy and Loss changes during 10 epochs.

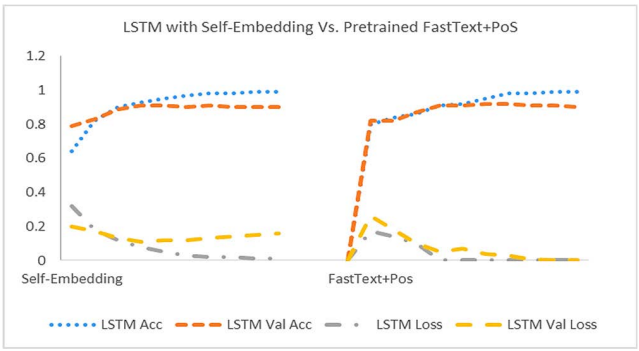

FIGURE 33. LSTM with Self-Embedding vs. LSTM with FastText+PoS training and validation accuracy and loss changes during 10 epochs.

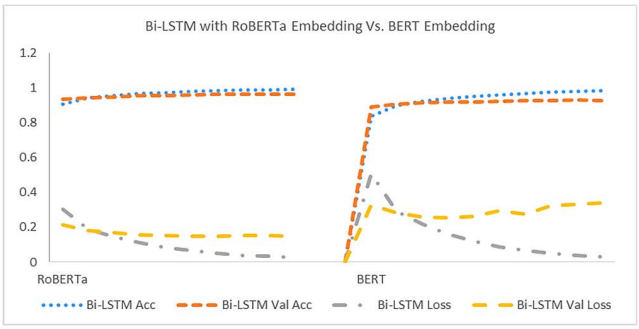

FIGURE 34. Bi-LSTM with BERT-Embedding vs. Bi-LSTM with roberta-embedding accuracy and loss changes during 10 epochs.

than the result from BERT embedding, as shown in Fig. 34. The implementation of transformer-based word embeddings with the Bi-LSTM deep neural network model has yielded the best performance for our classification problem, resulting in significantly improved accuracy of 93% and 96% for BERT+Bi-LSTM and RoBERTa+Bi-LSTM, respectively as shown in Fig. 35 to 37.

However, in order to achieve such higher accuracy, supplementary computation resources are needed to carry out the training processes for a longer period of time (240 and 275 minutes, respectively). After a thorough assessment of the relevant model literature; this study demonstrates that adopting Transformer-based pre-trained algorithms such as BERT and RoBERTa for NLP tasks achieves benchmark results and significantly improves the overall performance. According to our literature survey, the proposed implementation for both sentiment classification and geo-location prediction tasks demonstrates that our framework is competitive or outperforms state-of-the-art algorithms in several measures, as presented in Tables 8 and 10.

C. CROWD SITUATIONAL AWARENESS COMPUTATIONS AND VISUALIZATIONS

Hereby, we have employed the third proposed subsystem, the correlator, that was able to identify the crowd situational awareness by means of the computation of tem-

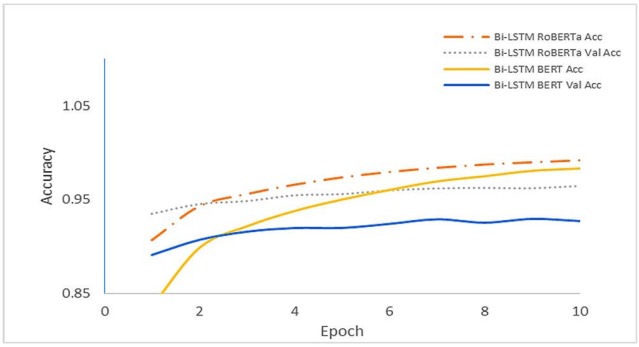

FIGURE 35. Bi-LSTM with bert and roberta classifier accuracy.

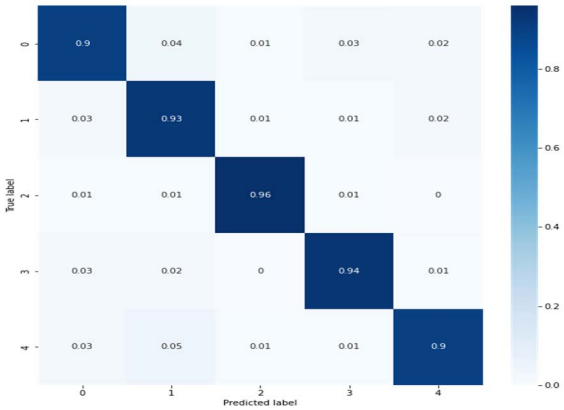

FIGURE 36. Bi-LSTM with BERT embedding classifier conf. matrix.

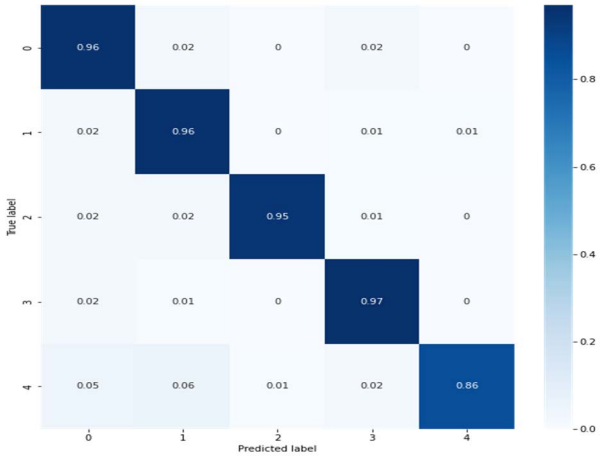

FIGURE 37. Bi-LSTM with roberta embedding classifier conf. matrix.

poral cross-correlation and time difference (lag or lead duration) between the emergence of negative sentiment tweets related to COVID-19 and COVID-19 daily confirmed active infections for the highly affected five countries as indicated in the previous section. Hereafter, some visualizations for Daily Tweets sentiment distribution according to the scraped dataset along with ''WHO'' reported COVID-19 daily active cases are presented for the countries under

TABLE 10. Proposed framework vs. literature results comparison.

Research Work	Accuracy
Proposed Framework, Sub-system 2: Geolocation Prediction	96%
W.Gad, 2021	85%
Zahid A Butt, 2020	62.5%
Sara Hasni, 2020	58.31%
Rhea Mahajan, 2021	92.6%

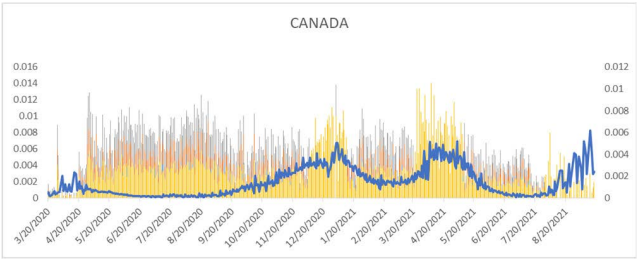

FIGURE 38. Tweets sentiment distribution along with ''WHO'' reported Covid-19 daily active cases for canada.

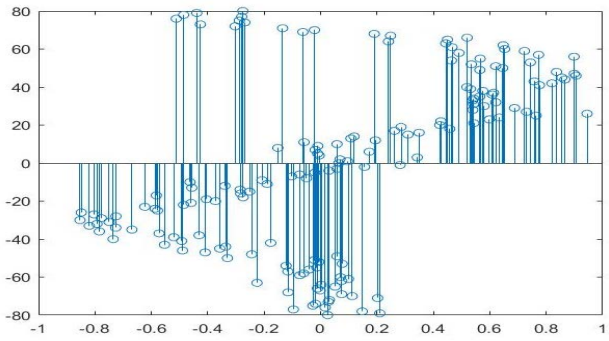

FIGURE 39. Cross-correlation coefficients with the corresponding time difference between negative tweets emerging and active cases reported counts in canada.

investigation, Canada, USA, India, Australia, and the UK, as shown in Fig. 38, Fig. 41, Fig. 44, Fig. 47 and Fig. 50, respectively. It shows the daily sentiment distribution according to the scraped dataset as the (yellow (negative), orange (positive), grey (neutral)) histogram while focusing on the negative sentiment tweets (yellow histogram), along with ''WHO'' reported Covid-19 daily active cases as the dark blue line representation. The sentiment analysis of the collected tweets revealed that from 40% to 50% of the tweeting users are classified as positive about the COVID-19 outbreak, 35% to 45% are neutral, and the remaining percentage is negative. This is aligned with our review of the literature for similar studies, which indicates that the majority of people viewed the epidemic positively and supported the government's or local authorities' actions.

Although the correlation between the two-time series is noticeable by the vision and can be expressed in non-numeric exact descriptions, we thought that exact numeric compar-

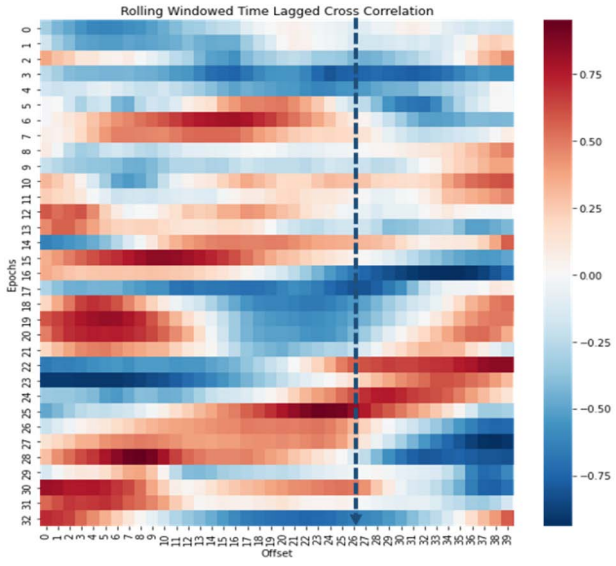

FIGURE 40. The heat map of the overlapping window time-lagged cross-correlation with the time-lag corresponds to the maximum coefficient.

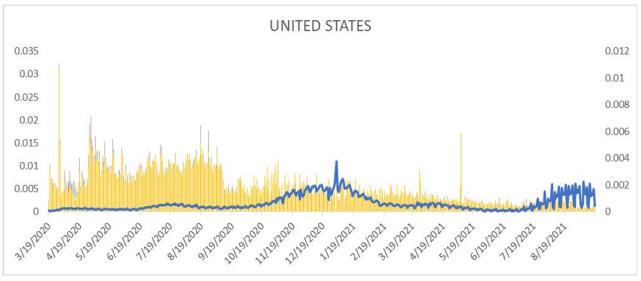

FIGURE 41. Tweets sentiment distribution according to the scraped dataset along with "WHO" reported Covid-19 daily active cases for USA.

isons could be of greater interest, especially in our case of tracking the progression of an infectious disease. So, we have developed a methodology for computing the correlation factors and the time difference between the two-time series (negative sentiment tweets and reported Covid-19 daily active cases) to be able to report how long the social media data was able to precede the actual reporting of the increase in active infectious cases. Our goal was to find the time difference that maximized the correlation coefficient between the two-time series.

The correlation factors and their corresponding time differences are shown in Fig. 39, Fig. 42, Fig. 45, Fig. 48, and Fig. 51, for Canada, the USA, India, Australia, and the UK, respectively. The highest cross-correlation and corresponding time difference for all countries under examination are given in Table 11, indicating that public awareness of the emerging infectious outbreak had a lead time difference ranging from 13 days to 26 days for all countries under investigation as illustrated in Fig. 40, Fig. 43, Fig. 46, Fig. 49, Fig. 52 presenting, the heat map of the overlapping window time-lagged cross-correlation with the time-lag (difference)

TABLE 11. Computed cross-correlation and time differences between negative tweets emergence and their corresponding COVID-19 active cases counts official reporting.

Country (Name)	Lead Time Difference (in Days)	Corresponding Cross-Correlation (%)
Canada	26	94.48
USA	21	94.2
India	18	96.80
Australia	23	91.47
UK	13	96.55

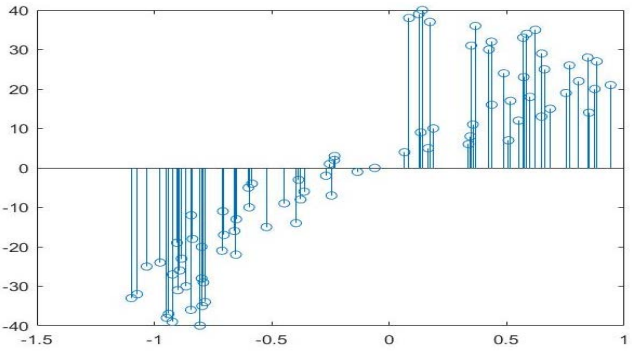

FIGURE 42. Cross-correlation with the corresponding between negative tweets emerging and active cases reported counts in USA.

corresponds to the maximum coefficient for Canada, USA, India, Australia, and the UK respectively, such that, if the computed cross-correlation was indicating that first time-series (volume/sentiment of tweets) leads the interaction with the second time-series (counts of daily active infections), it will be presented by red shades, higher cross-correlation coefficients, is presented by darker red shades. Time-lagged cross-correlations, or more specifically, overlapping (rolling) windowed time-lagged cross-correlations, are a great way to visualize the fine-grained dynamic interaction between two-time series, such as the leader-follower relationship and how it shifts over time, according to the presented visualizations of the cross-correlation between the two-time series under investigation. From these indicators, it is obvious that, on average, the designed social network analysis can provide a reliable early warning of an epidemic.

For validation purposes of twitter's lead time in response to the emerging COVID-19, a literature review was conducted to find that comparable values for time differences have been recognized in both the United States and the United Kingdom, according to the authors in [109] and [110] as presented in Table 12. These results show that crowd awareness in all countries predated the formal outbreak count reporting by governmental organizations. This component might be vital and invaluable in the early phases of outbreak monitoring. Many countries had difficulty identifying instances of the coronavirus in a timely manner since the disease was asymp-

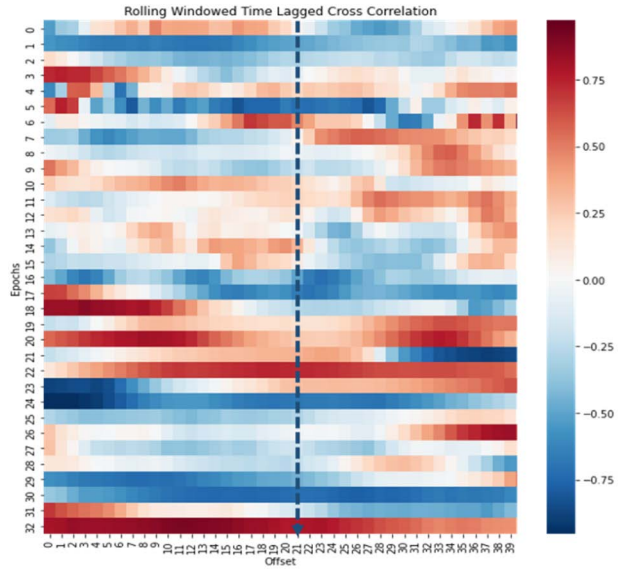

FIGURE 43. The heat map of the overlapping window time-lagged cross-correlation with the time-lag corresponds to the maximum coefficient.

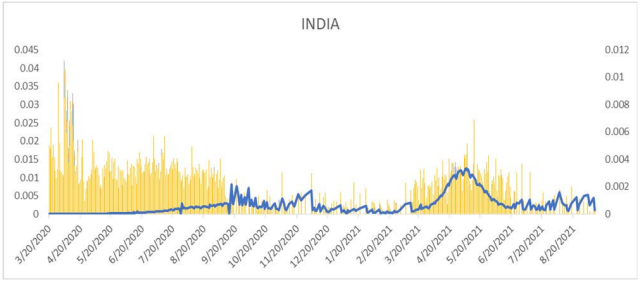

FIGURE 44. Tweets sentiment distribution according to the scraped dataset along with "WHO" reported Covid-19 daily active cases for India.

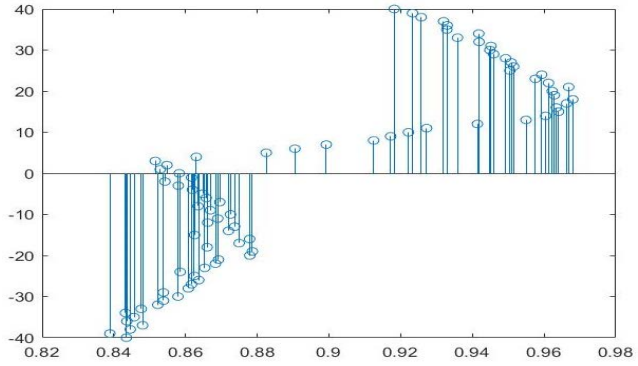

FIGURE 45. Cross-Correlation with the corresponding time difference between negative tweets and active cases reported counts in India.

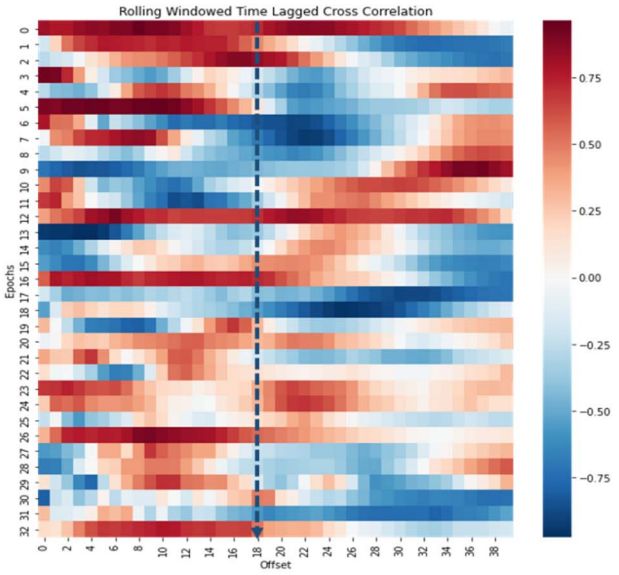

FIGURE 46. The heat map of the overlapping window time-lagged cross-correlation with the time-lag corresponds to the maximum coefficient.

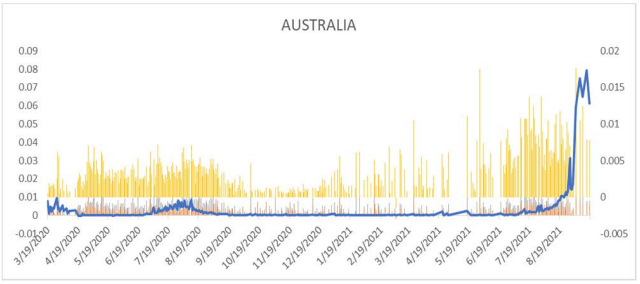

FIGURE 47. Tweets sentiment distribution according to the scraped dataset along with "WHO" reported Covid-19 daily active cases for Australia.

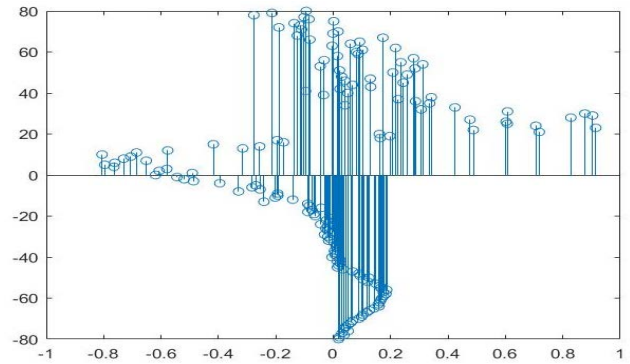

FIGURE 48. Cross-correlation with the corresponding time difference between negative tweets and active cases reported counts in Australia.

tomatic and/or its symptoms were difficult to differentiate from those of other diseases.

V. LIMITATIONS, STRENGTHS, AND COMPLEXITIES

A. LIMITATIONS

When used for research purposes, real-time posting on social media can be viewed as both a limitation and a strength. This is due to the rapidity with which social media content can evolve the potential existence of many outliers or fake-news spreaders. Authorities do not use social media platforms to disseminate formal news or control any trending topic or situation, increasing the likelihood of fake news spreading. Furthermore, social network platform owners imposed numerous restrictions on scraping and using publicly available data. Twitter, for example, only allows the scraping of

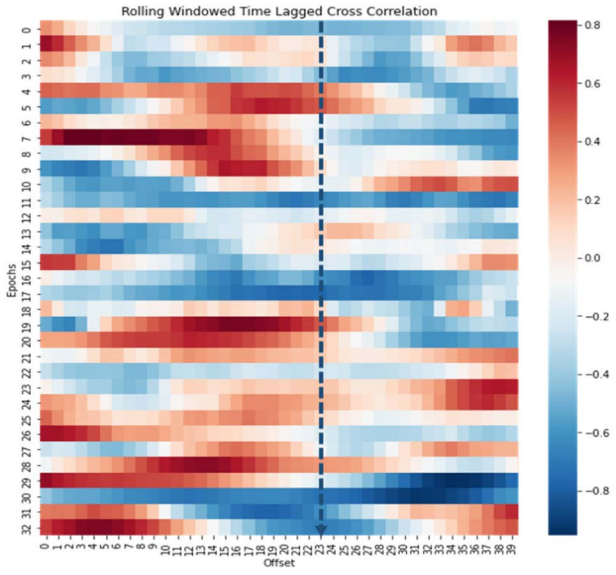

FIGURE 49. The heat map of the overlapping window time-lagged cross-correlation with the time-lag corresponds to the maximum coefficient.

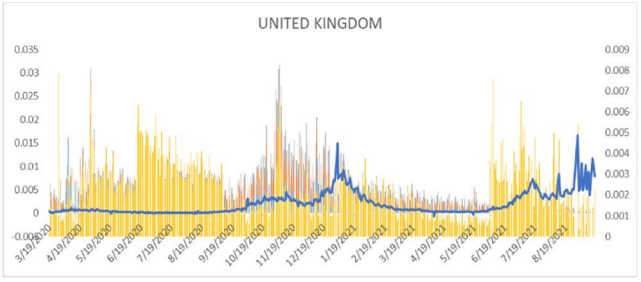

FIGURE 50. Tweets sentiment distribution according to the scraped dataset along with "WHO" reported Covid-19 daily active cases for the UK.

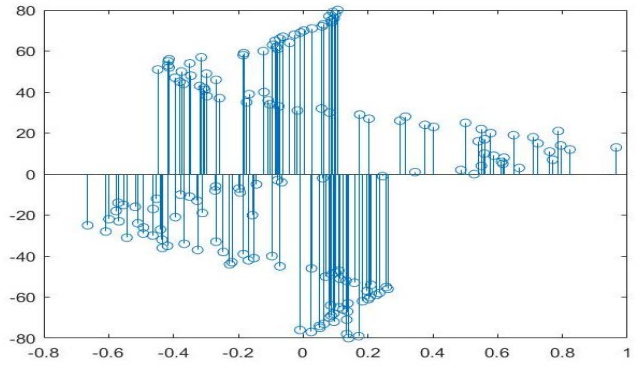

FIGURE 51. Cross-correlation with the corresponding time difference between negative tweets and active cases reported counts in the UK.

tweets no older than one week or the retrieval of a very limited number of tweets per minute, which may complicate the research process and cause unnecessary delays. In addition, current social network platforms only provide access to very limited demographic and geographic meta-data associations with user social interactions, which may limit the diversity of features required for many research problems.

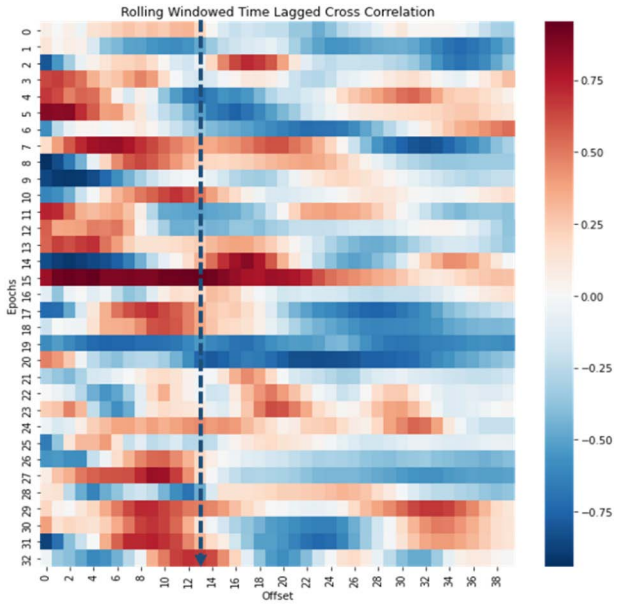

FIGURE 52. The heat map of the overlapping window time-lagged cross-correlation with the time-lag corresponds to the maximum coefficient.

TABLE 12. Proposed framework vs. literature results comparison.

Research Work	Lead Time
Sub-system 3: Public Awareness Quantification	21 (USA) 13 (UK)
Erfaneh Gharavi, 2020	5-19 (USA)
Lei Gao, 2021	16 (USA)
I Kit Cheng, 2021	6-27 (UK)

B. STRENGTH

Despite the limitations mentioned, social network data is, by definition, broadly accessible, real-time, and vast, with the potential to offer a variety of features related to human interactions on social network platforms. Furthermore, given the multilingual nature of social media data, the proposed methods in this study employ only textual data features and numerous text-dense representation algorithms to analyze the spread of an epidemic outbreak, which could be very advantageous to the scalability of the proposed framework to be applied to many other incidents or situations which might require analyzing non-English textual data. Despite the time complexity associated with modern artificial intelligence algorithms, they can be very useful in analyzing, assessing, and managing any unplanned situation, especially when empowered by vast amounts of data collected from social networks.

C. COMPLEXITY

Evaluating the complexity of a machine learning algorithm is not a simple task, given that it may be

implementation-dependent, the properties of the data may lead to other algorithms, and the training time is frequently dependent on certain parameters passed to the algorithm. When dealing with neural network-based learning algorithms, where multiple layers of processing in addition to large volumes of data inputs are required for the model training/fitting process, the same implication may be more apparent. For instance, the time complexity (Big-O measures) of $O(n^2f)$, $O(n^2fn_{trees})$, *and* $O(nfn_{trees})$ for Decision Trees, Random Forest, and Gradient Boosting, respectively, indicates that gradient boosting is less complex than the other two algorithms, where n; is the number of input instances, f; is the number of features, and n_{trees}; is the number of employed trees (if applicable). Regarding algorithms based on neural networks, the computational complexity for each layer is used as a metric to compare various structures' computational complexity. As per our proposed framework, the attention-based algorithms BERT and RoBERTa introduced in the Methods section are employed to build the embedding layer. The computational complexity per attention layer can be calculated as $O(l^2.d_{embed})$, where l is the input length, and d_{embed} is the embedding dimension. The computational complexity for a recurrent layer is computed as $O(l.d^2_{embed})$, whereas the computational complexity per convolutional layer is $O(k.l.d^2_{embed})$, where k is the kernel size of convolutions. For the majority of cutting-edge language models, the embedding size d_{model} ranges between 150 and 500, which must be greater than l, the average input length. As a result, we shall have $l^2{\cdot}d_{embed} \ll l{\cdot}d^2_{embed} \ll k{\cdot}l{\cdot}d^2_{embed}$. This relationship explains why the attention model operates significantly faster as an embedding layer encoder and why the recurrent network operates significantly faster than the convolutional neural network for the same process [111]. Hence, to determine the total time complexity per model for a neural network-based classifier, the complexity of each layer type must be calculated and added together. For example, the authors in [111] has computed the complexity of 3-layers MLP as; $C_{MLP} = n_s n_i n_1 + n_1 n_2 + n_2 n_3 + n_3 n_o$, where n_1, n_2, n_3 are dense layer neurons, n_s = input sequence size, n_i, n_o = input vector, output vector dimensions. Consequently, if this network is used with an embedding layer, the computed complexity must be increased to account for the added embedding layer complexity, etc. Since we employ pre-trained word embedding, which has the potential to reduce the time and space complexity required for the classification problem under study, it can be considered an extremely powerful feature. This is because pre-trained networks have learned the fundamental representations of data structures and can be trained on a small domain-specific dataset to provide accurate classification.

VI. CONCLUSION AND FUTURE STUDIES

Recently, the world health organization (WHO) announced that SARS-CoV-2 had infected over 10.5 million people and inflicted 5.5 million fatalities within six months from its first emergence. By the end of June 2022, the outbreak had affected 500 million people and inflicted 6.5 million people. The obvious discrepancy between the infected cases counts announced by all countries in June 2020, and June 2022 is aligned with WHO's declaration that the announced counts at the first emergence of the severely infectious COVID-19 don't accurately reflect the outbreak status due to many social, technical and financial factors. When COVID-19 emerged, the most pressing concern was how to mitigate the infection and protect billions of citizens globally without compromising the international economy. This was so challenging to all governments especially with the lack of accurate infection reporting. As a result, Many countries around the world have suffered economic disruptions due to national/regional lockdowns and quarantine that lasted days to months. Also, in response to the emergence of coronavirus disease, numerous governments, national healthcare organizations, and institutions have initiated a contact tracing network analysis over billions of GPS human mobility data points to monitor the evolution of the disease contact network. Alternatively, diverse studies have employed social network-based geo-positioning for tracing social contacts in order to examine the likely progression of the infection as a safer and more widely accepted alternative to sensor-based geo-positioning tracking applications. Our study investigates the importance of using social network data analysis and machine/deep learning-based algorithms to improve infectious disease awareness as a proactive infosurveillance and warning system for pandemic outbreaks. The proposed framework comprises three subsystems. The first subsystem includes data collection, integration, preprocessing, and hybrid sentiment analysis tools. The second subsystem comprised the feature extraction unit that identifies, selects, embeds, and balances feature vectors and the classifier fitting and training unit. The last subsystem measures temporal associations between pandemic-relevant social network activities and official infection reported counts in the most hazardous geolocations.

Given the multilingual nature of social media data, the proposed methods in this study analyze the spread of an epidemic outbreak using only textual data features and numerous text-dense representation algorithms. This could greatly improve the scalability of the proposed framework, allowing it to be applied to a wide range of other incidents or situations requiring the analysis of non-English textual data. We have employed deep learning methods to overcome many natural language processing (NLP) problems, including textual classification and sentiment analysis, by improving the text representation component and overall model design. We have assessed linear classifiers, MLPs, RNNs, and CNNs with word embedding algorithms like self-embedding, word2vec, GloVe, and FastText to improve location prediction accuracy. The proposed framework was developed and tested using static and real-time Twitter data from 577k geotagged COVID-19-related tweets. The experimental results of the first subsystem showed that the Decision Tree Classifier with Unigram+TF-IDF feature vectors outper-

formed other conventional models for sentiment classification with an accuracy of 94.3%. Additionally, this accuracy outperformed other systems discussed in the literature even with their employment with more advanced methods such as neural network-based classifiers. While for the second subsystem, we couldn't achieve a satisfactory performance by employing only conventional machine learning algorithms, such that we were able to achieve the best accuracy of 80% for Decision Tree Classifier with Unigram+TF-IDF. This drove us to the development of neural network-based classifiers such as Convolutional Neural Networks (CNN), Multi-Layer Perceptron (MLP), and Recurrent Neural Networks (RNN), empowered with word embedding algorithms such as self-embedding, word2vec, GloVe, and FastText, in addition to transformer-based word embedding such as BERT and RoBERTa. As per the evaluation of our customization of the mentioned methods, we found that the overall performance is highly improved when diverse linguistic features being considered. For instance, using MLP with a self-embedding layer, Word2Vec, and GloVe pre-trained word embedding resulted in an accuracy of 10%, 36%, and 32%, respectively. As well as, adding PoS tag one-hot encoding embedding increased validation accuracy from 36% to 89%. Bi-LSTM with RoBERTa word embedding has resulted in the best performance for the subsystem, as it was able to predict the top 5 hazardous countries with 96% accuracy, which outperformed relevant implementations as discussed in our literature. Additionally, The results of the third and last subsystem showed the framework's remarkable overall performance in assessing temporal associations between public awareness and outbreak status, while the outbreak's Twitter activities arise 13 to 26 days before the official nation-level reporting of confirmed infection counts for the geolocations under investigation. This result is aligned with several research studies discussed in our proposed literature for better performance and more automation for capturing the public awareness about an emerging situation while depending on computational methods instead of manual or trivial tools. Furthermore, it is evident that despite the time complexity of the modern AI algorithms, these can be useful for accurately analyzing, assessing, and managing unplanned situations, especially when empowered with massive amounts of data gathered from social networks. So, in our proposed framework, we considered the employment of pre-trained word embedding to be an extremely powerful feature because it has the potential effect of reducing the time and space complexity required for the classification problem under study. Accordingly, and as per our extensive study for various social network data analyses, and despite its limitations, social network data is widely available, real-time, and vast, and can provide a variety of features related to human interactions on social network platforms. So, we conclude that acquired information from tweets would be extremely useful if identified in a timely manner, and even more so if the location is known or precisely predicted. Thus, inferred Twitter user locations could assist in preventing the spread of a catastrophic epidemic outbreak, thereby saving lives. Such an approach is considered economically efficient to administer and treat infected patients if the disease is contained in its early stages of emergence. Ongoing and future work will assess other potentials of social network data, especially the users' connection networks, and demographics, to model the progression of an ongoing infectious outbreak using advanced techniques.

REFERENCES

[1] B. Murugesan, S. Karuppannan, A. T. Mengistie, M. Ranganathan, and G. Gopalakrishnan, "Distribution and trend analysis of COVID-19 in India: Geospatial approach," *J. Geographical Stud.*, vol. 4, no. 1, pp. 1–9, Apr. 2020, doi: 10.21523/gcj5.20040101.

[2] J. M. Read, J. R. E. Bridgen, D. A. T. Cummings, A. Ho, and C. P. Jewell, "Novel coronavirus 2019-nCoV (COVID-19): Early estimation of epidemiological parameters and epidemic size estimates," *Phil. Trans. R. Soc.*, 2021, Art. no. B3762020026520200265.

[3] B. Tang, X. Wang, Q. Li, N. L. Bragazzi, S. Tang, Y. Xiao, and J. Wu, "Estimation of the transmission risk of the 2019-nCoV and its implication for public health interventions," *J. Clin. Med.*, vol. 9, no. 2, p. 462, Feb. 2020, doi: 10.3390/jcm9020462.

[4] N. E. Huang and F. Qiao, "A data driven time-dependent transmission rate for tracking an epidemic: A case study of 2019-nCoV," *Sci. Bull.*, vol. 65, no. 6, pp. 425–427, Mar. 2020, doi: 10.1016/j.scib.2020.02.005.

[5] J. T. Wu, K. Leung, M. Bushman, N. Kishore, R. Niehus, P. M. de Salazar, B. J. Cowling, M. Lipsitch, and G. M. Leung, "Estimating clinical severity of COVID-19 from the transmission dynamics in Wuhan, China," *Nature Med.*, vol. 26, no. 4, pp. 506–510, Apr. 2020, doi: 10.1038/s41591-020-0822-7.

[6] M. U. G. Kraemer, C.-H. Yang, B. Gutierrez, C.-H. Wu, B. Klein, D. M. Pigott, L. du Plessis, N. R. Faria, R. Li, W. P. Hanage, J. S. Brownstein, M. Layan, A. Vespignani, H. Tian, C. Dye, O. G. Pybus, and S. V. Scarpino, "The effect of human mobility and control measures on the COVID-19 epidemic in China," *Science*, vol. 368, no. 6490, pp. 493–497, May 2020, doi: 10.1126/science.abb4218.

[7] Z. Yang, Z. Zeng, K. Wang, S.-S. Wong, W. Liang, M. Zanin, P. Liu, X. Cao, Z. Gao, and Z. Mai, "Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions," *J. Thoracic Disease*, vol. 12, no. 3, p. 165, 2020, doi: 10.21037/jtd.2020.02.64.

[8] J. S. Jia, X. Lu, Y. Yuan, G. Xu, J. Jia, and N. A. Christakis, "Population flow drives spatio-temporal distribution of COVID-19 in China," *Nature*, vol. 582, pp. 389–394, Apr. 2020, doi: 10.1038/s41586-020-2284-y.

[9] R. Fang, S. Pouyanfar, Y. Yang, S.-C. Chen, and S. S. Iyengar, "Computational health informatics in the big data age: A survey," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 1–36, Jul. 2016, doi: 10.1145/2932707.

[10] L. E. Charles-Smith, T. L. Reynolds, M. A. Cameron, M. Conway, E. H. Y. Lau, J. M. Olsen, J. A. Pavlin, M. Shigematsu, L. C. Streichert, K. J. Suda, and C. D. Corley, "Using social media for actionable disease surveillance and outbreak management: A systematic literature review," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0139701, doi: 10.1371/journal.pone.0139701.

[11] K. Wazny, "Applications of crowdsourcing in health: An overview," *J. Global Health*, vol. 8, no. 1, pp. 1–20, Jun. 2018, doi: 10.7189/jogh.08.010502.

[12] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: Promise and potential," *Health Inf. Sci. Syst.*, vol. 2, no. 1, pp. 1–10, Feb. 2014, doi: 10.1186/2047-2501-2-3.

[13] B. Batrinca and P. C. Treleaven, "Social media analytics: A survey of techniques, tools and platforms," *AI Soc.*, vol. 30, no. 1, pp. 89–116, Feb. 2015, doi: 10.1007/s00146-014-0549-4.

[14] J. Saura, P. Palos-Sanchez, and M. Rios Martin, "Attitudes expressed in online comments about environmental factors in the tourism sector: An exploratory study," *Int. J. Environ. Res. Public Health*, vol. 15, no. 3, p. 553, Mar. 2018, doi: 10.3390/ijerph15030553.

[15] J. R. Saura, A. Reyes-Menendez, and P. Palos-Sanchez, "Are black friday deals worth it? Mining Twitter users' sentiment and behavior response," *J. Open Innovation, Technol., Market, Complex.*, vol. 5, no. 3, p. 58, Aug. 2019, doi: 10.3390/joitmc5030058.

[16] F. Velicia-Martin, J.-P. Cabrera-Sanchez, E. Gil-Cordero, and P. R. Palos-Sanchez, "Researching COVID-19 tracing app acceptance: Incorporating theory from the technological acceptance model," *PeerJ Comput. Sci.*, vol. 7, p. e316, Jan. 2021, doi: 10.7717/peerj-cs.316.

[17] H. Wang, Z. Wang, Y. Dong, R. Chang, C. Xu, X. Yu, S. Zhang, L. Tsamlag, M. Shang, J. Huang, Y. Wang, G. Xu, T. Shen, X. Zhang, and Y. Cai, "Phase-adjusted estimation of the number of coronavirus disease 2019 cases in Wuhan, China," *Cell Discovery*, vol. 6, no. 1, pp. 1–8, Dec. 2020, doi: 10.1038/s41421-020-0148-0.

[18] S. Hsiang, D. Allen, S. Annan-Phan, K. Bell, I. Bolliger, T. Chong, H. Druckenmiller, L. Y. Huang, A. Hultgren, E. Krasovich, P. Lau, J. Lee, E. Rolf, J. Tseng, and T. Wu, "The effect of large-scale anti-contagion policies on the COVID-19 pandemic," *Nature*, vol. 584, no. 7820, pp. 262–267, Aug. 2020, doi: 10.1038/s41586-020-2404-8.

[19] H. T. N. Giang, J. Shah, T. H. Hung, A. Reda, L. N. Truong, and N. T. Huy, "The first Vietnamese case of COVID-19 acquired from China," *Lancet Infectious Diseases*, vol. 20, no. 4, pp. 408–409, 2020, doi: 10.1016/S1473-3099(20)30111-0.

[20] A. K. M. Chan, C. P. Nickson, J. W. Rudolph, A. Lee, and G. M. Joynt, "Social media for rapid knowledge dissemination: Early experience from the COVID-19 pandemic," *Anaesthesia*, vol. 75, no. 12, pp. 1579–1582, Dec. 2020, doi: 10.1111/anae.15057.

[21] M. A. Al-Garadi, M. S. Khan, K. D. Varathan, G. Mujtaba, and A. M. Al-Kabsi, "Using online social networks to track a pandemic: A systematic review," *J. Biomed. Informat.*, vol. 62, pp. 1–11, Aug. 2016, doi: 10.1016/j.jbi.2016.05.005.

[22] *COVID-19 Lockdowns*. Accessed: Feb. 2022. [Online]. Available: https://en.wikipedia.org/wiki/COVID-19_lockdowns

[23] *European Parliament BRIEFING EPRS | European Parliamentary Research Service Author: Costica Dumbrava Members' Research Service*, document Pe 649.384 EN Tracking Mobile Devices to Fight Coronavirus, Apr. 2020.

[24] M. Serafino, H. S. Monteiro, S. Luo, S. D. S. Reis, C. Igual, A. S. L. Neto, M. Travizano, J. S. Andrade, and H. A. Makse, "Digital contact tracing and network theory to stop the spread of COVID-19 using big-data on human mobility geolocalization," *PLOS Comput. Biol.*, vol. 18, no. 4, Apr. 2022, Art. no. e1009865, doi: 10.1371/journal.pcbi.1009865.

[25] I. Demir, H. Conover, W. F. Krajewski, B.-C. Seo, R. Goska, Y. He, M. F. McEniry, S. J. Graves, and W. Petersen, "Data-enabled field experiment planning, management, and research using cyberinfrastructure," *J. Hydrometeorol.*, vol. 16, no. 3, pp. 1155–1170, Jun. 2015, doi: 10.1175/JHM-D-14-0163.1.

[26] Z. Li, C. Wang, C. T. Emrich, and D. Guo, "A novel approach to leveraging social media for rapid flood mapping: A case study of the 2015 south Carolina floods," *Cartography Geographic Inf. Sci.*, vol. 45, no. 2, pp. 97–110, Mar. 2018.

[27] W. F. Krajewski, D. Ceynar, I. Demir, R. Goska, A. Kruger, C. Langel, R. Mantilla, J. Niemeier, F. Quintero, B.-C. Seo, S. J. Small, L. J. Weber, and N. C. Young, "Real-time flood forecasting and information system for the state of Iowa," *Bull. Amer. Meteorolog. Soc.*, vol. 98, no. 3, pp. 539–554, Mar. 2017, doi: 10.1175/BAMS-D-15-00243.1.

[28] G. Cervone, E. Sava, Q. Huang, E. Schnebele, J. Harrison, and N. Waters, "Using Twitter for tasking remote-sensing data collection and damage assessment: 2013 Boulder flood case study," *Int. J. Remote Sens.*, vol. 37, no. 1, pp. 100–124, Jan. 2016, doi: 10.1080/01431161.2015.1117684.

[29] J. P. de Albuquerque, B. Herfort, A. Brenning, and A. Zipf, "A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management," *Int. J. Geograph. Inf. Sci.*, vol. 29, no. 4, pp. 667–689, 2015, doi: 10.1080/13658816.2014.996567.

[30] Y. Feng and M. Sester, "Extraction of pluvial flood relevant volunteered geographic information (VGI) by deep learning from user generated texts and photos," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 2, p. 39, Jan. 2018, doi: 10.3390/ijgi7020039.

[31] C. Restrepo-Estrada, S. C. de Andrade, N. Abe, M. C. Fava, E. M. Mendiondo, and J. P. de Albuquerque, "Geo-social media as a proxy for hydrometeorological data for streamflow estimation and to improve flood monitoring," *Comput. Geosci.*, vol. 111, pp. 148–158, Feb. 2018, doi: 10.1016/j.cageo.2017.10.010.

[32] M. A. Sit, C. Koylu, and I. Demir, "Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and spatial analysis: A case study of hurricane Irma," *Int. J. Digit. Earth*, vol. 12, no. 11, pp. 1205–1229, Nov. 2019, doi: 10.1080/17538947.2018.1563219.

[33] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[34] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Social Netw. Anal. Mining*, vol. 11, no. 1, pp. 1–19, Dec. 2021, doi: 10.1007/s13278-021-00776-6.

[35] S. F. Tsao, H. Chen, T. Tisseverasinghe, Y. Yang, L. Li, and Z. A. Butt, "What social media told us in the time of COVID-19: A scoping review," *Lancet Digit. Health*, vol. 3, no. 3, pp. e175–e194, 2021, doi: 10.1016/S2589-7500(20)30315-0.

[36] N. C. Dang, M. N. Moreno-García, and F. D. la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, Mar. 2020, doi: 10.3390/electronics9030483.

[37] M. R. Huq, A. Ali, and A. Rahman, "Sentiment analysis on Twitter data using KNN and SVM," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 19–25, 2017.

[38] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Informaion Retr. (SIGIR)*, 2003, pp. 235–242.

[39] S. Soni and A. Sharaff, "Sentiment analysis of customer reviews based on hidden Markov model," in *Proc. Int. Conf. Adv. Res. Comput. Sci. Eng. Technol. (ICARCSET)*, 2015, pp. 1–5, doi: 10.1145/2743065.2743077.

[40] X. Zhang and X. Zheng, "Comparison of text sentiment analysis based on machine learning," in *Proc. 15th Int. Symp. Parallel Distrib. Comput. (ISPDC)*, Jul. 2016, pp. 230–233, doi: 10.1109/ISPDC.2016.39.

[41] V. Malik and A. Kumar, "Communication. Sentiment analysis of Twitter data using naive Bayes algorithm," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 6, no. 4, pp. 144–149, Apr. 2018.

[42] N. Mehra, S. Khandelwal, and P. Patel, *Sentiment Identification Using Maximum Entropy Analysis of Movie Reviews*. Stanford, CA, USA: Stanford Univ., 2002.

[43] H. Wu, J. Li, and J. Xie, "Maximum entropy-based sentiment analysis of online product reviews in Chinese," in *Automotive, Mechanical and Electrical Engineering*. Boca Raton, FL, USA: CRC Press, 2017, pp. 559–562.

[44] S. Naz and S. Parveen, "Twitter sentiment analysis using convolutional neural network," in *Proc. Int. Conf. Smart Data Intell. (ICSMDI)*, vol. 25, May 2021, p. 9, doi: 10.2139/ssrn.3852906.

[45] M. Bates, "Tracking disease: Digital epidemiology offers new promise in predicting outbreaks," *IEEE Pulse*, vol. 8, no. 1, pp. 18–22, Jan./Feb. 2017, doi: 10.1109/MPUL.2016.2627238.

[46] R. Khan, "Social media analysis with AI: Sentiment analysis techniques for the analysis of Twitter COVID-19 data," *Crit. Rev.* vol. 7, no. 9, pp. 2761–2774, 2020.

[47] K. H. Manguri, R. N. Ramadhan, and P. R. M. Amin, "Twitter sentiment analysis on worldwide COVID-19 outbreaks," *Kurdistan J. Appl. Res.*, vol. 5, no. 3, pp. 54–65, May 2020, doi: 10.24017/covid.8.

[48] K. Nargund and S. Natarajan, "Public health allergy surveillance using micro-blogs," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2016, pp. 1429–1433, doi: 10.1109/ICACCI.2016.7732248.

[49] K. Lee, A. Agrawal, and A. Choudhary, "Forecasting influenza levels using real-time social media streams," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Aug. 2017, pp. 409–414, doi: 10.1109/ICHI.2017.68.

[50] R. A. Calix, R. Gupta, M. Gupta, and K. Jiang, "Deep gramulator: Improving precision in the classification of personal health-experience tweets with deep learning," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2017, pp. 1154–1159, doi: 10.1109/BIBM.2017.8217820.

[51] L. Li, "Can social media data be utilized to enhance early warning: Retrospective analysis of the U.S. COVID-19 pandemic," *medRxiv*, Jan. 2021, doi: 10.1101/2021.04.11.21255285.

[52] L. Sousa, R. de Mello, D. Cedrim, A. Garcia, P. Missier, A. Ucha, A. Oliveira, and A. Romanovsky, "VazaDengue: An information system for preventing and combating mosquito-borne diseases with social networks," *Inf. Syst.*, vol. 75, pp. 26–42, Jun. 2018, doi: 10.1016/j.is.2018.02.003.

[53] A. Abd-Alrazaq, D. Alhuwail, M. Househ, M. Hamdi, and Z. Shah, "Top concerns of tweeters during the COVID-19 pandemic: Infoveillance study," *J. Med. Internet Res.*, vol. 22, pp. 1–9, Apr. 2020.
[54] R. Chandrasekaran, V. Mehta, T. Valkunde, and E. Moustakas, "Topics, trends, and sentiments of tweets about the COVID-19 pandemic: Temporal infoveillance study," *J. Med. Internet Res.*, vol. 22, no. 10, Oct. 2020, Art. no. e22624, doi: 10.2196/22624.
[55] P. Kaila and A. V. Prasad, "Informational flow on Twitter-corona virus outbreak-top modelling approach," *Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 3, pp. 128–134, 2020. [Online]. Available: https://ssrn.com/abstract=3565169
[56] M. Mansoor, K. Gurumurthy, and V. R B. Prasad, "Global sentiment analysis of COVID-19 tweets over time," 2020, *arXiv:2010.14234*.
[57] A. S. Imran, S. M. Daudpota, Z. Kastrati, and R. Batra, "Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets," *IEEE Access*, vol. 8, pp. 181074–181090, 2020, doi: 10.1109/ACCESS.2020.3027350.
[58] C. Li and A. Sun, "Fine-grained location extraction from tweets with temporal awareness," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2014, pp. 43–52, doi: 10.1145/2600428.2609582.
[59] A. Schulz, "A multi-indicator approach for geolocalization of tweets," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 7, no. 1, 2013, pp. 573–582. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14396
[60] Y. Ikawa, M. Enoki, and M. Tatsubori, "Location inference using microblog messages," in *Proc. 21st Int. Conf. Companion World Wide Web (WWW) Companion*, 2012, pp. 687–690, doi: 10.1145/2187980.2188181.
[61] B. Han, P. Cook, and T. Baldwin, "Geolocation prediction in social media data by finding location indicative words," in *Proc. COLING*, 2012, pp. 1–18.
[62] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson, "The where in the tweet," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 2473–2476, doi: 10.1145/2063576.2063995.
[63] S. Chandra, L. Khan, and F. B. Muhaya, "Estimating Twitter user location using social Interactions—A content based approach," in *Proc. IEEE 3rd Int. Conf. Privacy, Secur., Risk Trust IEEE 3rd Int. Conf. Social Comput.*, Oct. 2011, pp. 838–843, doi: 10.1109/PASSAT/SocialCom.2011.120.
[64] H.-w. Chang, D. Lee, M. Eltaher, and J. Lee, "@Phillies tweeting from philly? Predicting Twitter user locations with spatial word usage," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 111–118, doi: 10.1109/ASONAM.2012.29.
[65] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: A content-based approach to geo-locating Twitter users," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 759–768, doi: 10.1145/1871437.1871535.
[66] S. Abrol and L. Khan, "Tweethood: Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, Aug. 2010, pp. 153–160, doi: 10.1109/SocialCom.2010.30.
[67] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: Improving geographical prediction with social and spatial proximity," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 61–70, doi: 10.1145/1772690.1772698.
[68] F. Bouillot, P. Poncelet, and M. Roche, "How and why exploit tweet's location information?" in *Proc. AGILE Int. Conf. Geographic Inf. Sci.* Cham, Switzerland: Springer, 2012, pp. 1–5.
[69] D. Jurgens, "That's what friends are for: Inferring location in online social media platforms based on social relationships," in *Proc. Int. AAAI Conf. Web Social Media*. vol. 7, no. 1, 2013, pp. 273–282.
[70] Y. Takhteyev, A. Gruzd, and B. Wellman, "Geography of Twitter networks," *Social Netw.*, vol. 34, no. 1, pp. 73–81, 2012, doi: 10.1016/j.socnet.2011.05.006.
[71] M. Gaman and R. T. Ionescu, "Combining deep learning and string kernels for the localization of Swiss German tweets," 2020, *arXiv:2010.03614*.
[72] A. Schulz, "A multi-indicator approach for geolocalization of tweets," in *Proc. Int. AAAI Conf. Web Social Media*. vol. 7, no. 1, 2013, pp. 573–582. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14396
[73] B. Han, P. Cook, and T. Baldwin, "A stacking-based approach to Twitter user geolocation prediction," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2013, pp. 7–12.
[74] B. Han, P. Cook, and T. Baldwin, "Text-based Twitter user geolocation prediction," in *Proc. JAIR*, vol. 49, 2014, pp. 451–500.
[75] J. McGee, J. Caverlee, and Z. Cheng, "Location prediction in social media based on tie strength," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2013, pp. 459–468, doi: 10.1145/2505515.2505544.
[76] D. Jurgens, "That's what friends are for: Inferring location in online social media platforms based on social relationships," in *Proc. Int. AAAI Conf. Web Social Media*. vol. 7, no. 1, 2013, pp. 273–282. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14399
[77] T. Zhong, T. Wang, F. Zhou, G. Trajcevski, K. Zhang, and Y. Yang, "Interpreting Twitter user geolocation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 853–859, doi: 10.18653/v1/2020.acl-main.79.
[78] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang, "Towards social user profiling: Unified and discriminative influence model for inferring home locations," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 1023–1031, doi: 10.1145/2339530.2339692.
[79] R. Mahajan and V. Mansotra, "Predicting geolocation of tweets: Using combination of CNN and BiLSTM," *Data Sci. Eng.*, vol. 6, no. 4, pp. 402–410, Dec. 2021, doi: 10.1007/s41019-021-00165-1.
[80] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Surv. Rev.*, vol. 23, no. 176, pp. 88–93, Apr. 1975, doi: 10.1179/sre.1975.23.176.88.
[81] A. Mostafa, W. Gad, T. Abdelkader, and N. Badr, "Pre-HLSA: Predicting home location for Twitter users based on sentimental analysis," *Ain Shams Eng. J.*, vol. 13, no. 1, Jan. 2022, Art. no. 101501, doi: 10.1016/j.asej.2021.05.015.
[82] S. Hasni and S. Faiz, "Word embeddings and deep learning for location prediction: Tracking coronavirus from British and American tweets," *Social Netw. Anal. Mining*, vol. 11, no. 1, pp. 1–20, Dec. 2021, doi: 10.1007/s13278-021-00777-5.
[83] B. Huang and K. M. Carley, "On predicting geolocation of tweets using convolutional neural networks," in *Proc. Int. Conf. Social Comput., Behav.-Cultural Modeling Predict. Behav. Represent. Modeling Simulation*. Cham, Switzerland: Springer, 2017, pp. 281–291, doi: 10.1007/978-3-319-60240-0_34.
[84] B. Huang and K. M. Carley, "A hierarchical location prediction neural network for Twitter user geolocation," 2019, *arXiv:1910.12941*.
[85] U. Qazi, "GeoCoV19: A dataset of hundreds of millions of multilingual COVID-19 tweets with location information," *SIGSPATIAL Special*, vol. 12, no. 1, pp. 6–15, Jun. 2020.
[86] R. Lamsal, "Coronavirus (COVID-19) geo-tagged tweets dataset," *IEEE DataPort*, Apr. 2020. Accessed: Dec. 5, 2021. [Online]. Available: https://ieee-dataport.org/open-access/coronavirus-covid-19-geo-tagged-tweets-dataset, doi: 10.21227/fpsb-jz61.
[87] A. Hirose, Eds., *Neural Information Processing: 23rd International Conference ICONIP*. Kyoto, Japan: Springer, Oct. 2016.
[88] S. Sarlis and I. Maglogiannis, "On the reusability of sentiment analysis datasets in applications with dissimilar contexts," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.* Cham, Switzerland: Springer, 2020, pp. 409–418, doi: 10.1007/978-3-030-49161-1_34.
[89] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[90] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[91] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, Dec. 2016, doi: 10.1162/tacl_a_00051.
[92] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[93] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[94] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, Sep. 2016, doi: 10.1016/j.eswa.2016.03.028.
[95] A. Dey, M. Jenamani, and J. J. Thakkar, "Lexical TF-IDF: An n-gram feature space for cross-domain classification of sentiment reviews," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.* Cham, Switzerland: Springer, 2017, pp. 380–386, doi: 10.1007/978-3-319-69900-4_48.

[96] B. Wilson, "An overview of word2vec," in *Proc. Presentation File, Berlin ML Meetup*, 2014, pp. 1–28.
[97] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, Jan. 2022, doi: 10.1109/TKDE.2020.2981314.
[98] Y. Kang, Z. Cai, C.-W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: A literature review," *J. Manage. Anal.*, vol. 7, no. 2, pp. 139–172, Apr. 2020, doi: 10.1080/23270012.2020.1756939.
[99] N. Gupta and R. Agrawal, "Application and techniques of opinion mining," in *Hybrid Computational Intelligence for Pattern Analysis and Understanding, Hybrid Computational Intelligence*. New York, NY, USA: Academic, 2020, pp. 1–23.
[100] O. Miguel-Hurtado, R. Guest, S. V. Stevenage, G. J. Neil, and S. Black, "Comparing machine learning classifiers and Linear/Logistic regression to explore the relationship between hand dimensions and demographic characteristics," *PloS One*, vol. 11, no. 11, Nov. 2016, Art. no. e0165521.
[101] S. Gan, S. Shao, L. Chen, L. Yu, and L. Jiang, "Adapting hidden naive Bayes for text classification," *Mathematics*, vol. 9, no. 19, p. 2378, Sep. 2021.
[102] J. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[103] R. Aggrawal and S. Pal, "Prediction of heart disease with different attributes combination by data mining algorithms," in *Computational Vision and Bio-Inspired Computing*. Springer, Singapore, 2021, pp. 469–482, doi: 10.1007/978-981-33-6862-0_38.
[104] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 355–370, Feb. 2015, doi: 10.1007/s13042-015-0328-7.
[105] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[106] S. M. Boker, "Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series," *Psychol. Methods*, vol. 7, no. 3, p. 338, 2002, doi: 10.1037/1082-989X.7.3.338.
[107] J. H. Cheong, "Synchronized affect in shared experiences strengthens social connection," *PsyArXiv*, Jan. 2020, doi: 10.31234/osf.io/bd9wn.
[108] N. G. E.-D. Saad, S. Ghoniemy, H. Faheem, and N. A. Seada, "An evaluation of time series-based modeling and forecasting of infectious diseases progression using statistical versus compartmental methods," in *Proc. 5th Int. Conf. Comput. Informat. (ICCI)*, Mar. 2022, pp. 263–273, doi: 10.1109/ICCI54321.2022.9756060.
[109] E. Gharavi, N. Nazemi, and F. Dadgostari, "Early outbreak detection for proactive crisis management using Twitter data: COVID-19 a case study in the U.S.," 2020, *arXiv:2005.00475*.
[110] I. K. Cheng, J. Heyl, N. Lad, G. Facini, and Z. Grout, "Evaluation of Twitter data for an emerging crisis: An application to the first wave of COVID-19 in the U.K.," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, Dec. 2021, doi: 10.1038/s41598-021-98396-9.
[111] P. J. Freire, Y. Osadchuk, B. Spinnler, A. Napoli, W. Schairer, N. Costa, J. E. Prilepsky, and S. K. Turitsyn, "Performance versus complexity study of neural network equalizers in coherent optical systems," *J. Lightw. Technol.*, vol. 39, no. 19, pp. 6085–6096, Oct. 2021.

NOHA GAMAL received the bachelor's degree in electrical engineering from the Faculty of Engineering, Alexandria University, in 2001, and the master's degree in communication and information technology from Nile University, Cairo, Egypt, in 2015. She is currently pursuing the Ph.D. degree with the Faculty of Information and Computer Science, Ain Shams University, Cairo. From 2002 to 2014, she was a communication and information engineer and a communication manager in many national and international companies and organizations. Since 2014, she has been a lecturer with many national universities.

SAMY GHONIEMY (Member, IEEE) is a Professor of Computer Systems at the British University in Egypt (BUE), with significant industrial and governmental consultation experience. Ghoniemy obtained his B.Sc. and M.Sc. in Communications and Computer Engineering in 1990 and 1996. Ghoniemy served as a member of a national committee preparing the Egyptian Artificial Intelligence Strategy. He is interested in collaborative work that addresses artificial intelligence, satellite and onboard computer design, optical and tactical networks; IoT and smart campus networks; cognitive science; semantic networks for cancer disease detection; and dynamic graphs for predicting and controlling epidemic and endemic diseases. Dr. Ghoniemy authored and co-authored more than 100 research papers published in IEEE, Springer, and Elsevier. He is now an active expert in the National Academy of Scientific Research and Technology (ASRT) and the Egyptian Space Agency (EgSA).

HOSSAM M. FAHEEM received the B.Sc. (Hons.) and M.Sc. degrees in computer engineering and science from the Faculty of Electronic Engineering, Egypt, in 1992 and 1995, respectively, and the Ph.D. degree from the Department of Computers and Systems Engineering, Faculty of Engineering, Ain Shams University, Egypt, in 2000. He is currently a Full Professor of computer systems with the Faculty of Computer and Information Sciences, Ain Shams University. He has participated in, supervised, and managed many types of research and research projects in multi-agent based systems, parallel processing, high-performance computing networking, and security.

NOHA A. SEADA received the B.Sc., M.S., and Ph.D. degrees in scientific computing from the Faculty of Computer and Information, Ain Shams University, Cairo, Egypt, in 2004, 2009, and 2017, respectively. She is currently a Lecturer and a Postdoctoral Researcher with the Computer Systems Department, Faculty of Computer and Information, Ain Shams University. Her research interests include machine learning, computer vision, deep learning, medical image analysis, and artificial intelligence. She received the Google Anita Borg Memorial Scholarship for Europe, the Middle East, and Africa, in 2011. She was awarded a scholarship in 2013 to conduct part of her Ph.D. research at Waterloo University's Vision and the Image Processing Laboratory in Canada.