Computers, Materials & Continua
DOI:10.32604/cmc.2022.018972
Article

Tech Science Press

An Ensemble Learning Based Approach for Detecting and Tracking COVID19 Rumors

Sultan Noman Qasem[1,2], Mohammed Al-Sarem[3,4] and Faisal Saeed[3,*]

[1]Computer Science Department, College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 11432, Saudi Arabia
[2]Computer Science Department, Faculty of Applied Science, Taiz University, Taiz, 6803, Yemen
[3]College of Computer Science and Engineering, Taibah University, Medina, Saudi Arabia
[4]Information System Department, Saba'a Region University, Mareeb, Yemen
[*]Corresponding Author: Faisal Saeed. Email: fsaeed@taibahu.edu.sa

Received: 28 March 2021; Accepted: 07 May 2021

Abstract: Rumors regarding epidemic diseases such as COVID 19, medicines and treatments, diagnostic methods and public emergencies can have harmful impacts on health and political, social and other aspects of people's lives, especially during emergency situations and health crises. With huge amounts of content being posted to social media every second during these situations, it becomes very difficult to detect fake news (rumors) that poses threats to the stability and sustainability of the healthcare sector. A rumor is defined as a statement for which truthfulness has not been verified. During COVID 19, people found difficulty in obtaining the most truthful news easily because of the huge amount of unverified information on social media. Several methods have been applied for detecting rumors and tracking their sources for COVID 19-related information. However, very few studies have been conducted for this purpose for the Arabic language, which has unique characteristics. Therefore, this paper proposes a comprehensive approach which includes two phases: detection and tracking. In the detection phase of the study carried out, several standalone and ensemble machine learning methods were applied on the Arcov-19 dataset. A new detection model was used which combined two models: The Genetic Algorithm Based Support Vector Machine (that works on users' and tweets' features) and the stacking ensemble method (that works on tweets' texts). In the tracking phase, several similarity-based techniques were used to obtain the top 1% of similar tweets to a target tweet/post, which helped to find the source of the rumors. The experiments showed interesting results in terms of accuracy, precision, recall and F1-Score for rumor detection (the accuracy reached 92.63%), and showed interesting findings in the tracking phase, in terms of ROUGE L precision, recall and F1-Score for similarity techniques.

Keywords: Rumor detection; rumor tracking; similarity techniques; COVID-19; social media analytics

1 Introduction

Social media are commonly used to spread the messages, alerts and other news worldwide and have currently become one of the main news sources, rather than other, more traditional, platforms. In addition, huge advancements in technology, such as the use of smart phones, makes it easy to spread information very fast, regardless of its credibility [1]. It is difficult to verify the veracity of information spread on social media, especially during a disaster or similar crisis [2]. The information that is usually spread by non-credible sources is called a rumor and can be spread by a huge number of people on social media in a short time [3]. Rumors can cause various effects on economic, political and other aspects of the global society and their transmission has an increasing substantial impact on human lives and social stability [4,5]. During these situations, governments must play an important role in order to maintain sustainable market development [6–8]. For instance, during COVID-19, people in many countries felt scared once the World Health Organization declared it a pandemic and therefore many rumors spread on social media about specific drugs which can prevent the disease or reduce the infection, causing high demand for these drugs which affected the sustainability of the entire healthcare market [8].

Several studies have focused on the impact of rumors during disasters and crises. For instance, Kim and Kim [9] investigated the factors influencing the rumors associated with the Fukushima nuclear accident. In further studies, on the COVID-19 pandemic [10,11], they also investigated the effects of health beliefs on preventive behaviors and analyzed the belief structure of COVID-19 rumors, which caused what is known as an infodemic. Zhang et al. [12] investigated how health-related rumors mislead the perception of people during a public health emergency. According to [13], the efficient and effective detection of rumors is highly important in order to minimize this harmful impact, and that the detection task is not simple. This makes the work on automatic identification of rumors from social media a hot research topic [1]. One of the issues that makes rumor detection more challenging is the labeling task, which is time-consuming and requires rigorous labor work [3]. The other common challenges are feature extraction from a given dataset, retrieving the data from the sources and database bias and quality [14].

Several methods have been applied for detecting rumors from social media, including supervised, unsupervised and hybrid machine learning approaches [1]. For instance, Alkhodair et al. in [13], introduced a method and trained a recurrent neural network to detect rumors related to breaking news that are propagated on social media.. Their experiments used a real-life dataset and applied the proposed method for cross-topic early rumor detection. They found that this method outperforms the previous methods in terms of several metrics, including precision and recall. In addition, Wu et al. [3] investigated the issue from an important angle by studying the ability of knowledge learned from old data to detect new rumors. Using real-world datasets, they found that the applied methods were effective. Another study [15] introduced a model based on Recurrent Neural Networks (RNN) for rumor identification that depends on learning the sequential posts by utilizing its temporal hidden representation. Wu et al. in [16] proposed a hybrid model for rumor detection based on a convolutional neural network (CNN), where the layer of the CNN uses the recurrent structure. In addition, Roy et al. [17] introduced an architecture for rumor detection based on ensemble learning. In order to classify the rumors, they used CNN and Bi-directional Long- Short-term Memory (BI-LSTM). The experimental outcomes of these methods were passed to a multilayered perceptron method for performing the final classification. However, this proposed ensemble architecture obtained an accuracy of only 44.87%.

Rumor detection and source identification in a social network are considered very important tasks for controlling the diffusion of misinformation and have recently gained the attention of researchers in social media analytics area [18]. There are some websites that make tracking simple, such as snopes.com and emergent.info, which manually collect stories and classify them as rumors; however, the task of automatically tracking the source of rumors still challenging. Detecting the accurate sources of rumors is also considered a challenging issue, because of the dynamic evolution of the network of social media. Several methods have been used to investigate this tracking issue; for instance, Shao et al. [19] developed a system to collect, detect and analyze online misinformation for tracking purposes. They collected the data from news websites and social media. They found that rumors are controlled by active users, whereas fact-checking is a more grass-roots activity. In addition, graph-based methods have been applied for tracking the spread of rumors. According to Shelke et al. [18], the main steps for detecting the source of rumors in a Twitter social network start by identifying the rumor and collecting its dataset, which includes sender, receiver and sent post. The data should then be preprocessed in order to remove stop words, hashtags, URLs, and other unnecessary information, and then the data should be annotated. After that, the rumor's propagation is constructed and the appropriate diffusion model selected. Finally, the sources are classified based on metrics of source detection, and the outcomes are evaluated using actual and estimated sources. Some studies worked with rumor source identification as a tree-like network [19–22]. Yu et al. [23] applied a finite graph and use the message-passing approach for source detection, to reduce the search of vertices for estimating the maximum likelihood. In another approach, Xu et al. [24] proposed a source detection method by applying sensor nodes in the network that do not use the rumor's text. The authors' of [25] introduced a rumor source detection method in a temporal network based on the Susceptible-Infected-Recovered model (SIR). In addition, other approaches were used for detecting the source of rumors on social media such as a query-based approach [26], anti-rumor-based approach [27], ranking-based approach [28], community-based approach [29] and approximation-based approach [30].

Rumors become more harmful when they are related to the spread of health misinformation. Several research efforts have investigated detection and tracking of health-related rumors. For instance, in [31] the authors conducted a study to examine the people who are spreading health-related rumors, such as publicizing ineffective cancer treatments. The study involved 4,212 Twitter users and 139 ineffective "treatments". Features such as user writing style and sentiment were used with a classification method that obtained 90% accuracy. In addition, [32] reported a tool for tracking the health-related rumors on Twitter that worked on tweets related to the Zika outbreak. More than 13 million tweets were collected and the tool pipeline, which included health professionals, crowdsourcing and machine learning, provided a method to detect the health-related rumors. In addition, identifying the rumor early during a disaster is considered very important and helps to avoid many health issues. Mondal et al. [33] introduced a probabilistic model, in which the prominent features of rumor propagation are combined. The content-based analysis was then performed to guarantee the contribution of the extracted tweets in terms of the probability of being a rumor. According to [34], several methods worked in detecting rumors from social media using machine learning and other techniques. However, they found that few studies have focused on detecting health-related rumors in Arabic language. Thus, they introduced a process of building a health-related rumors dataset and applied several machine learning techniques to detect health-related rumors in the Arabic language. The applied techniques detected the rumors with an accuracy of 83.50%. During the COVID-19 pandemic, the issue of spreading rumors has become more harmful and affects many aspects of life. Spreading these rumors covering the healthy behaviors and publicizing wrong practices can lead to increasing the rate of spreading the virus. Therefore, advanced technologies such as data mining methods are needed to detect the online posts that include rumors from social media [35]. Few studies have addressed this important issue, especially in the Arabic language. In this regard, Haouari et al. [36] built the ArCOV19-Rumors dataset using Arabic language for COVID19 misinformation detection in Twitter. However, detecting and tracking the rumors related to COVID19 using Arabic language still a big challenge and requires more research.

In this paper, a comprehensive approach for detecting and tracking the source of rumors related to COVID 19 in the Arabic language is proposed. In the rumor-detecting phase, several machine learning methods including Linear Regression, K-nearest Neighbor, Decision Tree (CART), Support Vector Machine and Naïve Bayes (Bernoulli) were applied and investigated. In addition to individual classifiers, several ensemble learning methods were applied such as Random Forest, AdaBoost, Bagging, Extra-Trees and Stacking that worked on the tweets' texts. In addition, the Genetic Algorithm-based Support Vector Machine model (GA-SVM) was applied on the user's and tweet's features. The proposed detection model then combined the ensemble model and the GA-SVM model that obtained the best performance and used these in the second phase, rumor tracking. In previous studies, the Bayesian network-based similarity method was used to identify the rumors of texts and predict the characters of users more accurately and effectively [37]. In this proposed approach, several similarity measures such as Cosine, Jaccard, and Chebyshev were used to compare the target query with the detected rumors to obtain its source.

The organization of this paper is as follows. Section 2 covers the research background. Section 3 describes the materials and methods used in this study, including dataset description, data preprocessing and the proposed model. Section 4 presents the details of the experimental results and the discussion. Section 5 compares the performance of the similarity techniques used in tracking the source of rumors. Section 6 concludes the paper by highlighting the main contributions and suggests future work.

2 Related Studies

This section reviews the methods for detecting rumors and the source of rumors and applications related to rumor detection and tracking. Most methods for rumor detection currently use supervised learning. The most popular methods are content-based algorithms. Content-based approaches identify misinformation or false news according to the texts' or pictures' truthfulness. These works presume the material in various types of rumors (or news) varies in some quantifiable manner. In each article with a particular subject matter relating to health-related reporting, the refined features inspired by the theory of graphics and paradigms of social factors were used [38]. Rumors sometimes include images. Thus, Vishwakarma et al. [39] suggest a platform-independent validation system to verify news by analyzing the authenticity of photographic information. The news is described by the paradigm in four stages, the first stage is extracting the text from the pictures; then naming the entities from the text in the second stage; and the third stage involves *scraping* the web for associated information according to the extracted entities and then classification occurs in the final stage.

Some researchers immediately extracted features by including deep learning algorithms to reduce the deficiencies of conventional approaches based on content. For instance, Kaliyar et al. [40] suggest a false news content-based identification, FNDNet, based on a deep convolutional neural network. The algorithm that they propose is developed to study discriminative features for detection of false media automatically through multiple hidden layers built into a deep neural network. Zhang et al. [41] suggest a multi-layer structural neural network Auto-Encoder (AE) automated detection system for rumor detection. In addition, multiple thresholds to enable rumor identification have been suggested to self-adapt. A novel automated rumor detection system based on a long-/short-term memory classifier is proposed in [42]. The algorithm applied in this work not only obtained greater precision, F1 score and accuracy, but also had low false positivity. Ajao et al. [43] proposed a system for identification and classification of false news from Twitter messages, based on mixed convolutional neural networks and long-term recurrent neural networks. Their system helped improve efficiency as it did not need the vast number of training data characteristics required by deep learning models. The analysis of counterfeit news distributed over multiple social media sites poses new problems that render previously applied algorithms inefficient or inaccurate. To address these issues, [44] evaluated four common machine learning algorithms to verify their utility separately, in terms of identification and classification of false news. Many emerging approaches actually classify rumors focused only on language knowledge, without taking temporal dynamics and transmission patterns into account. Another study, by Wu et al. [45] proposed a new way of creating a spreadsheet using a Twitter spreadsheet. A gated graphical neural network algorithm was then used, which can produce powerful images for each propagation graph node.

Identifying the origins of rumors in social media is also critical. This is required to mitigate the problems created by the dissemination of rumor throughout society. The consequences of pervasive disinformation on individuals and culture can be unacceptable, negative, and even destructive [46]. The distribution of knowledge on social media has led to several developments in research, such as identification of disinformation or gossip, social bust awareness, tracking of the propagation of false news, estimation of potential diffusion and rumor detection. To counteract these effects, researchers have performed numerous experiments, including psycholinguistic analysis, computer training and deep learning methods from multiple perspectives. The propagation of misinformation on a network poses a variety of threats, including fear of an epidemic infection among the public and wrong decisions by authorities in a crisis. Thus, it is really important to avoid and monitor the rapid dissemination of rumors in social networks. Early rumor detection [47], verification of the veracity of rumors [48] or misinformation and recognition of the rumor's source [49] will monitor the rumor propagation in a network. Non-credible material spreads quickly online through social networks. It is extremely difficult to detect the origins of misinformation in a fast and accurate way, due to the complicated distribution process, credible evidence and complex network adjustments in the social network. Most recently, a few social media tools have been developed for rumor identification and analysis. However, these instruments do not track or control the development of diffusion, and are completely unable to detect any particular origins.

The study by Louni et al. [50] presents a two-phase algorithm which is used for finding the source. The volatility in social networks is quantified using a probabilistic weighted graph. Recently, several algorithms have suggested that clusters in complex networks can be calculated. Thus, the first phase of the process consists of clustering and deciding the most possible cluster, using the Louvain clustering algorithm. The first set of algorithms is based on the division of graphs until the required number of clusters is reached. It has been suggested that the algorithm can classify groups through node similarities. Maryam et al. [51] proposed a heuristic-based approach for identifying the doubtful origin of the deceptive dissemination, while Ji et al. [52] developed a systematic frame and methodology for the identification of multiple sources based on estimators developed for the identification of a single source. Their model is developed to predict when infections will begin, at distinct periods.

The identification of sources is critical in different fields of operation Because of its wide variety of uses, major advances in the identification of origins have been observed in the last two decades. Significant research has been conducted into sources in a range of application areas, such as healthcare (the first patient to be discovered to monitor an influenza pandemic) [53], surveillance (computer virus sources) [54], and wide interconnected networks (wireless sensor network gas leak source [55], e-mail network source [56], dynamic network propagating sources [57] and social network rumor disinformation sources [23,27]).

3 Materials and Methods

To identify the source of rumors, a two-phase approach for rumor detection and tracking was proposed. In the first phase, a detection process was conducted (detecting phase) in which we aimed to classify the collected posts as rumor or non-rumor. Once the post was classified, the set of rumor posts was fed into the second phase (tracking phase). In the detection phase, we conducted extensive experiments with conventional and ensemble machine learning models on a collection of posts (dataset). First, a set of conventional classifiers was applied and tested, which were (i) Logistic Regression (LR), (ii) K-Nearest Neighbor (KNN), (iii) Classification and Regression Tree (CART), (iv) Support Vector Machine (SVM), and (v) Bernoulli Naïve Bayes (NB). A set of ensemble classifiers was then investigated, which were (i) Random Forest (RF), (ii) AdaBoost, (iii) Bagging, (iv) ExtraTree, (v) stacking-based ensemble classifiers, and (vi) the Stochastic Gradient Descent (SGD) classifier. The best performing method here was the stacking-based ensemble model that worked on the tweet's texts. In addition, we applied a Genetic Algorithm-based Support Vector Machine model (GA-SVM) on the users' and tweets' features. The proposed method then combined the two models: the stacking-based ensemble model and applied genetic algorithm-based Support Vector Machine model to obtain the best detection for COVID-19 rumors in Arabic.

In the tracking phase, if the target post was classified as a rumor, the set of predicted rumors from the detection phase along with this target rumor post were fed into the similarity techniques process to identify the most similar posts (rumors) that could be considered as the source for this post (rumor). Three similarity techniques were used: (i) Cosine-based Similarity, (ii) Jaccard-based Similarity, and (iii) Chebyshev Distance. We also investigated the effect of using the Arabic GLoVe [58] pre-trained word embedding vector on the overall similarity techniques.

3.1 Database Description

The dataset used is available publicly [59]. The data repository was organized to serve analysis of several social network sites. The data available on the "tweet verification" sub-directory were used. This sub-directory holds the contents of all the annotated tweets. The directory also contains information about the propagation of tweets. Therefore, there are two components in this dataset:

- Tweets file as a tab-separated file. The file stores a t_i tweet as $t_i = \langle t_{id}, t_\iota \rangle$, where t_{id}: a tweet ID and t_ι: veracity labels ($t_\iota \in \{True, False\}$).
- Propagation networks: contains the IDs of retweets and conversational threads for the tweets.

Since the Tweets file sorts only the tweets' IDs, the Hydrator[1] was used to collect tweets using these IDs, and it was found that a set of tweets was missing[2]. At the end, a set of tweet-based features and user-based features was obtained. The obtained tweets' metadata were then concatenated with the veracity labels found in the Tweets file. The process of concatenation is described in Algorithm 1. The overall data statistics can be found in Tab. 1.

Algorithm 1: Dealing with Missing Tweets

```
  Input: TweetFile $F \langle t_{id}, t_\iota \rangle$, $t_{id}$: TweetsID, $t_\iota$: VeracityLabel
  Output: Final Dataset $\mathfrak{D}$
// Initialization of Tweet Text empty list
    $df \leftarrow$ hydrator_dataset(); // obtained metadata using hydrator tool
    $temp \leftarrow \{\}$;
for each $t \in df$.index do
    for $x$ in $F$.index do // Tweets File
        if $F[t_{id}][x] == df\ [t_{id}][t]$ then
            $temp$.append($F[t_{id}][x]$);
            $i \leftarrow i + 1$;
$\mathfrak{D} = Zip(F, df, temp)$;
return $\mathfrak{D}$;
```

3.2 Data Preprocessing

The Hydrator tool produces 34 linguistic and user features from a tweet. For the proposed model, we used the tweet's texts (text of the posted tweet by the user) to be trained by individual and ensemble models and the users' and tweets' features to be trained by GA-SVM model. Since the dataset used is slightly unbalanced (see Tab. 1), the Synthetic Minority Oversampling Technique (SMOTE) was performed in order to augment the number of rumors (Fig. 1).

Table 1: Data statistics of ArCOV-19 dataset

Features	Tweets' Statistics
Original dataset size	3612
% of missing data	455 (12.59%)
Obtained data $\mathfrak{D}$	3157 Rumors 1480 (46.87%), 1677 non-Rumors (53.12%)
Data frame	27 January 2020–15 May 2020
Language	Arabic

The extracted texts, including rumor and non-rumor tweets, were then moved to the next stage, where several preprocessing techniques were applied:

- hashtags were removed and the word after each tag was kept,
- URL removal and whitespace removal,

[1] https://github.com/DocNow/hydrator

[2] Out of 3612 tweets listed in the original Tweets file, only 3157 tweets were obtaind.

- the word “COVID-19” was replaced with “19-كوفيد,”
- non-Arabic character removal,
- Normalization[3],
- Stemming, using ISRI stemmer and lemmatization[4].

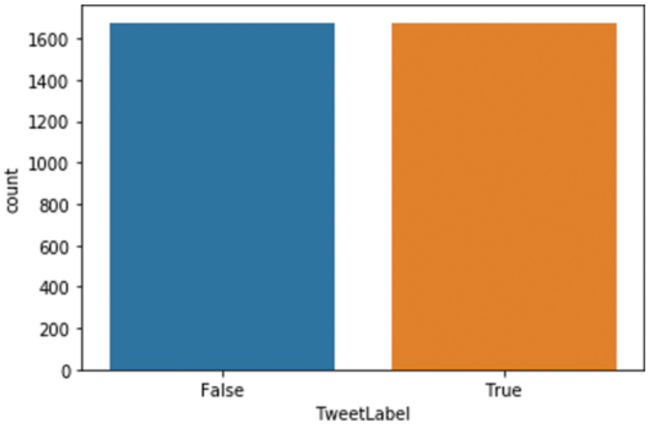

Figure 1: Dataset Size after applying SMOTE technique

The cleaned texts were then used by different standalone classifiers to classify rumor and non-rumor tweets. However, before feeding the text into the classifier, the tf/idf technique was used as a tokenization method, as we are concerned here about the representation and detection of rumors’ texts, while the tweets’ texts were represented using the Arabic GLoVe pre-trained word embedding vector, as we are concerned here about the meaning of the tweets and the similarity with the query post (tweet). The detailed description of tweet representation is in the subsection below, while the detailed description of the users’ and tweets’ features is presented in Section 3.3.3.

Tweet Representation in the Detection phase

In the detection phase, we converted the collection of the preprocessed tweets to the matrix of the tf/idf feature using n-gram. The lower and upper boundaries of the n-gram were one and three, respectively. This means that we capture at the same time the unigram, bigram, and trigram. This allows us to catch phrases such as كورونا - Corona, “19-كوفيد” - Covid-19 and “مرض كورونا المستجد” “novel Coronavirus”. Therefore, given two tweets t_i and t_j, where the word count in each tweet is $\ell \leq 240$, tf-idf with n-gram is represented as:

$tf\_idf = \{tf_{idf_{n=1}}, tf_{idf_{n=2}}, tf_{idf_{n=3}}\}$, where

$tf_{idf_{n=1}}$ is the tf_idf matrix with respect to the unigram, $tf_{idf_{n=2}}$ is the tf_idf matrix with respect to the bigram, $tf_{idf_{n=3}}$ is the tf_idf matrix with respect to the trigram, m is the terms in

[3] PyArabic library is used: https://github.com/linuxscout/pyarabic

[4] NLTK library

each tweet, and n is the number of tweets in the collection. Thus, the final matrix of the tf_idf feature is presented as follows:

$$tf_{idf} = \begin{bmatrix} e^p_{11} & e^p_{21} & \cdots & e^p_{m1} \\ e^p_{12} & e^p_{22} & \vdots & e^p_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ e^p_{1n} & e^p_{2n} & & e^p_{mn} \end{bmatrix}, \text{ where}$$

$$e^p_{ij} = log\left(1 + tf_{w,t}\right) \times log_{10}(N/df_w),\ p \in [1, 2, 3]$$

3.3 Detection Phase

In this phase, several models that work on the tweets' texts were applied, including standalone machine learning models and ensemble-based machine learning models,. In addition, the GA-SVM model was applied, which worked on both users' and tweets' features. The proposed model was then applied, which combined the stacking ensemble model and the GA-SVM model to obtain the best detection rate for COVID-19 rumors in Arabic.

3.3.1 Model 1-Standalone Machine Learning Models

As stated earlier, five machine learning models were used, namely LR, KNN, CART, SVM and NB. These models were used later as base classifiers for ensemble methods. The base classifier was selected based on its ability to deal with high dimensional data, its performance when the dataset size is increased, and its sensitivity to noise data [60]. The detailed model configurations and hyper-parameter settings are presented in Tab. 2. These models work on the tweets' texts.

Table 2: Hyper-parameter settings for each of the classifiers

Classifier name	Hyper-parameters used
Linear regression	C = 1.0, intercept_scaling = 1, l1_ratio = None, max_iter = 100, penalty = 'l2', solver = 'lbfgs', tol= 0.0001, verbose = 0
K-nearest Neighbor	leaf_size = 30, distance = 'minkowski', #neighbors = 5, p = 2, weights = 'uniform'
Decision Tree (CART)	criterion = 'gini', min_samples_leaf = 1, min_samples_split = 2
Support Vector Machine	C = 1000, gamma= 0.001, kernel = 'rbf'
Naïve Bayes (Bernoulli)	alpha = 1.0, binarize = 0.0, class_prior = None, fit_prior = True

3.3.2 Model 2- Ensemble-based Machine Learning Models

In recent years, ensemble learning has gained more interest [61]. The ensemble-based model improves the overall classification performance by fusing the output of a set of base classifiers [60]. Given a pool of Mbase classifiers, some classifiers usually perform better than others. Thus, finding a way to combine them tends to be more accurate than working with each classifier separately. In literature, ensemble learning models can be either homogeneous ensembles such bagging [62], boosting [63], random forest [64] and a SGD classifier [65] or heterogeneous ensembles such as stacking [66]. The detailed model configurations and hyper-parameter settings can be seen in Tab. 3.

Table 3: Detailed model configurations and hyper-parameter settings for ensemble learning methods

Classifier name	Base classifier	Hyper-parameters
RF	CART	#estimators = 100, criterion = ‘gini’, bootstrap = True, ccp_alpha = 0.0, min_samples_leaf: 5, min_samples_split: 12.
SGD	_	loss = ‘hinge’, max_iter = 1000, n_iter_no_change = 5, penalty = ‘l2’
AdaBoost	SVM,	Probability = True, kernel = ‘linear’
	CART, LR	*
Bagging	CART	max_samples = 1.0, n_estimators = 10, best_estmator*
	LR	max_samples = 1.0, n_estimators = 10, best_estmator*
	Knn	leaf_size = 5, n_neighbors = 7_neighbor, p = 1, distance = minkowski
ExtraTree	Default	Default
Stacking	Level0**, level1 = LR	*
	Level0**, level1 = SVM	*
	Level0**, level1 = CART	*

Notes:
*The same configuration settings as in Tab. 2, **[RF, KNN, CART, SVM, NB].

3.3.3 Model 3- Genetic Algorithm-Based Support Vector Machine Model

In addition to the tweets’ texts, a set of user-based features and tweet-based features are extracted. The user-based features are: (i) number of user’s friends, (ii) number of followers, (iii) number of favorites accounts that user likes, (iv) verified user or not, and (v) number of public lists. The tweet-based features are: (vi) retweet count, (vii) favorite count, and (viii) sensitive content. The complete description of the features is shown in Tab. 4.

Since the extracted features have different variances and some of them have missing values, standardization of data and missing data handling techniques were performed. For tuning the proposed classifier, the Support Vector Machine was trained using the aforementioned extracted features with the following parameter settings of GA, as shown in Tab. 5. The detailed results of the different classifiers are shown in Section 4.

3.3.4 Model 4-The Proposed Model: Combined Stacking Classifier (LR) and GA-SVM

The outcomes of the second and third models were combined by concatenating them to form a new training set, which was later fed to the GA-SVM classifier that was trained and tested using k-fold cross validation. The proposed model is illustrated in Fig. 2.

Table 4: List of user-based features with their type and description

Type	Feature	Description
User-based	Number of user's friends	The number of users this account is following (AKA their "followings")
	Number of followers	The number of followers this account currently has
	Number of favorite accounts	The number of tweets this user has liked in the account's lifetime
	Number of public lists	The number of public lists that this user is a member of
	Verified user	Binary indicator: Twitter account is verified or not
Tweet-based	Retweet count	Number of times this tweet has been retweeted.
	Favorite count	Approximately how many times this tweet has been liked by Twitter users.
	Possible sensitivecontent	An indicator that the URL contained in the tweet may contain content or media identified as sensitive content

Table 5: Parameter settings of GA used with the GA-based SVM

Parameter	Value
Generations	10
Population size	24
Mutation rate	0.02
Crossover rate	0.5
Early stop	12

3.4 Tracking Phase

To track the source of any rumor (tweet), we passed the output of the detection phase into the tracking phase. Since we were concern only with the rumors' texts here, each tweet that was predicted as a rumor was represented using the Arabic GLoVe pre-trained word embedding vector (as shown in Fig. 3). The next sections describe how this process was done.

3.4.1 Tweet Representation at Tracking phase

The target of the tracking phase is to find those previous rumors that share similar concepts and meanings to a specific tweet. Thus, the Arabic GLoVe pre-trained word embedding vector was used to represent each tweet in a fixed dimension of a real-value vector. GloVe word embedding was used to map each word to a 50-dimensional vector. We averaged the 50 dimensions of each word, w_i. Thus, the complete word embedded of a tweet t_i is mapped from 50 dimensions to 1 dimension as follows:

$$t_i = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1m} \\ e_{21} & e_{22} & \vdots & e_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ e_{n1} & e_{n2} & & e_{nm} \end{bmatrix} \Longrightarrow \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}, \text{ where}$$

$$\mu_i = \frac{1}{m} \sum_{k=1}^{m} e_{ik}$$

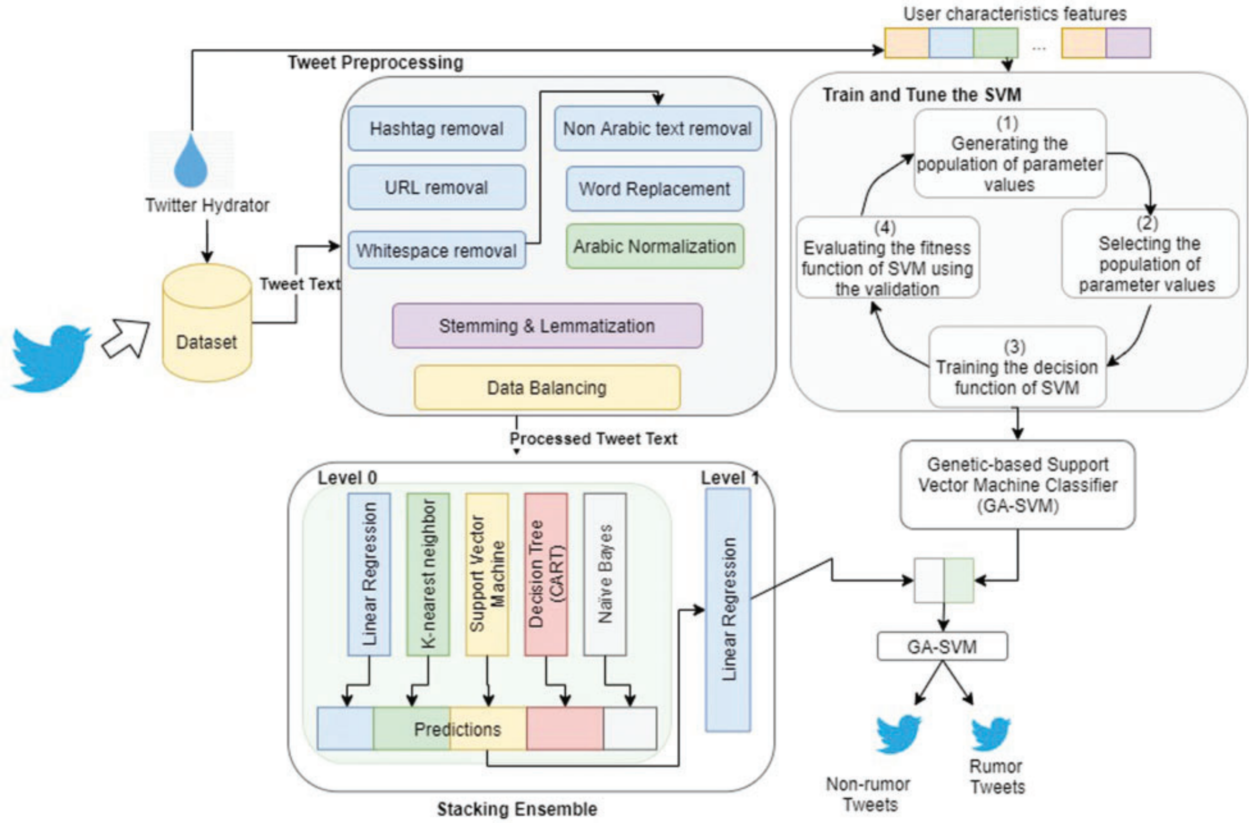

Figure 2: Combined GA-based SVM with stacking ensemble model

3.4.2 Similarity Measures

With the final average vector of any tweet, finding the similarity between tweet t_i and t_j can be conducted in several ways. Three similarity techniques were used: (i) Cosine-based similarity, (ii) Jaccard-based similarity, and (iii) Chebyshev distance. Algorithm 2 is used to reduce the number of operations needed.

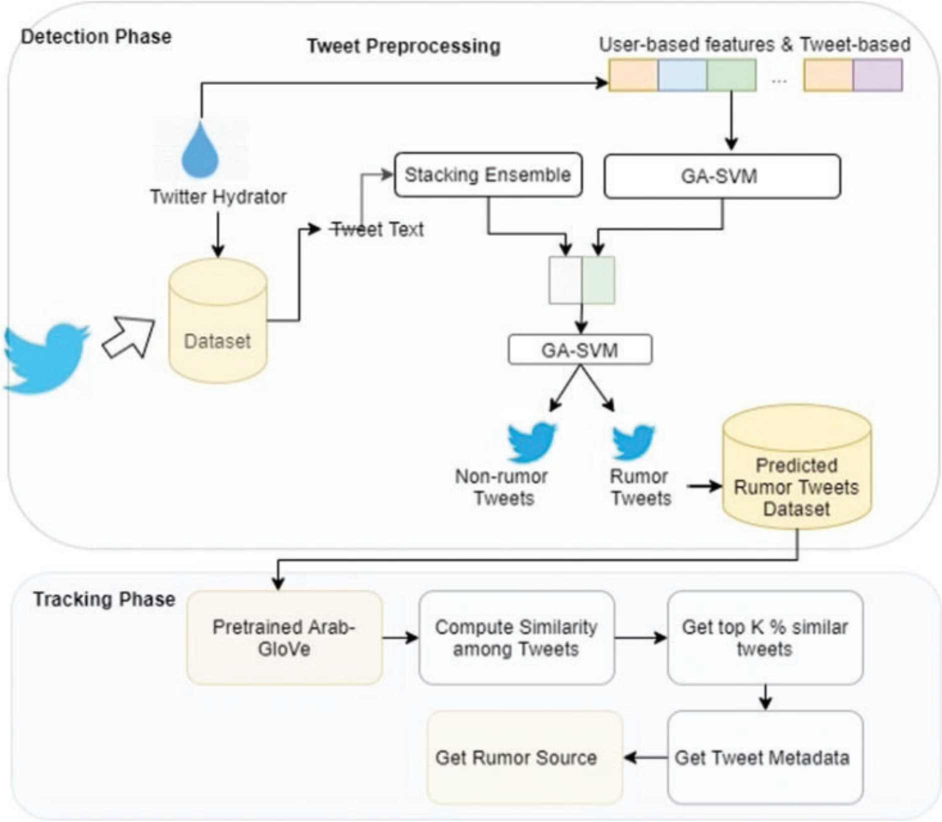

Figure 3: Rumor tracking phase

Algorithm 2: Reduction of Source Set( )

```
Input: RumorTweets $R_t$, Timestamp of each tweet $T$, StartingTweet $t^*$
Output: Candidate set of source rumors $R_t^{'}|R_t^{'} \in R_t$
$R_t^{'} \leftarrow []$;
for $i$ in $R_t$ do
    if $T(R_i)$ is ealier than $T(t^*)$ OR $t^*$ is retweet of $R_i$ Then
        $R_t^{'}$.append($R_i$);
return CandidateSet $R_t^{'}$
```

3.4.3 Source Detection

Assume a tweet t_i is the target rumor post which was classified by a classifier M as a rumor. To find the most potential sources of rumors, the reduction algorithm is first executed to obtain the candidate set $R_t^{'}$, as shown in Algorithm 2. The similarity between t_i and each tweet in the candidateSet $R_t^{'}$ is then computed and only the top K components of *scores* set are returned, as shown in Algorithm 3.

```
Algorithm 3: Similarity Between Two Tweets( )
  Input: RumorTweets $R_t$, Timestamp of each tweet $T$, StartingTweet $t^*$
  Output: SimilarityScore: $Scores[]$
$Scores \leftarrow []$
  $RumorsList \leftarrow []$, $Score = 0$
for $i$ in $R_t$ do
  $RumorsList = ReductionOfSourceSet()$
for each $i \in RumorsList$ do
  $Score \leftarrow ComputeSimilarity(t^*, R_i)$ // Apply Cosine/Jaccard/Chebyshev Similarity
  $Scores$.append ($Score$)
return Top $K$components of $Scores$
```

3.4.4 Evaluation Metrics

The performance of the proposed detection models was evaluated using accuracy, recall, precision and F1-score. Since the repeated stratified k-fold cross validation was used, each evaluation matric was averaged and the standard error computed. Accuracy (Acc), recall (R), precision (P), and F1-score (F1) were computed as shown in Eqs. (1)–(4), respectively.

- Accuracy: the ratio of accurately predicted tweets, either as rumors or not ($TP + TN$), to the total data set $\mathfrak{D}$.
- Recall: the number of accurately predicted rumor tweets (TP) to the total number of actual rumor tweets ($TP + FN$).
- Precision: the number of accurately predicted rumor tweets (TP) to the total number of predicted rumor tweets ($TP + FP$)
- F1-score: the harmonic mean between precision and recall, which gives the balanced evaluation between both precision and recall.

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{|\mathfrak{D}|} \tag{1}$$

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

$$P = \frac{TP}{\mathrm{TP} \ + \ \mathrm{FP}} \tag{3}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{4}$$

The average value of any evaluation metric and standard error are computed as in Eqs. (5) and (6) respectively.

$$\mu = \frac{1}{n} \sum_{i=1}^{10} x_i, \text{ where } \mathrm{x_i} : \text{ evaluation matric within } \mathrm{i}^{\mathrm{th}} \text{ fold} \tag{5}$$

$$\sigma_x = \frac{\sigma}{\sqrt{n}}, \text{where } \mathrm{n} = 3, \ \sigma \text{ is the standard deviation of the population} \tag{6}$$

The ROUGE values (ROUGE L precision, recall and F-Measure) were used for the similarity approaches used for evaluating the performance of the proposed tracking algorithm.

4 Results and Discussion

This section discusses the results of the proposed two-phase rumor detection and tracking approach. The experimental part of this work was performed on Python 3.8 with Windows 10 operating system. We used sklearn 0.22.2 as the main Python package for implementing the classifiers. The classifiers were evaluated using a repeated stratified k-fold cross validator with 10 folds, which needed to be repeated 3 times. The same preprocessing steps were used for each classifier to make a fair comparison between classifiers.

4.1 Results of Model 1- Standalone Machine Learning Models

We started the experiments by passing the cleaned and tokenized tweets' texts to five standalone machine learning classifiers. As stated earlier, the repeated stratified k-fold cross validator with 10 folds was used. The performance of the classifiers was reported in terms of average accuracy, average recall, average precision, and average f-score.

Out of the five classifiers used, support vector machine, Bernoulli naïve Bayes and linear regression obtained a very similar performance, where they reached an average accuracy of 90.7%, 90.5%, and 90.3% respectively. The worst performance was that of the K-NN classifier, which achieved an average accuracy of 69.4%, average of precision of 0.626, and average F-score of 0.760. In terms of recall, K-NN achieved a recall of 0.917, as shown in bold in Tab. 6.

Table 6: Performance of base classifiers

Classifier name	Results							
	Accuracy		Recall		Precision		F-score	
	μ	σ	μ	σ	μ	σ	μ	σ
Linear Regression	0.903	0.017	0.879	0.031	0.924	0.020	0.901	0.018
K-nearest Neighbor	0.694	0.021	**0.917**	0.013	0.626	0.016	0.760	0.013
Decision Tree (CART)	0.860	0.017	0.810	0.031	0.906	0.021	0.854	0.018
Support Vector Machine	0.907	0.012	0.892	0.023	0.921	0.020	0.906	0.013
Naïve Bayes (Bernoulli)	0.905	0.014	0.876	0.027	0.931	0.020	0.903	0.015

In addition, the box plots shown in Fig. 4 indicate that SVM gives a robust performance compared to other classifiers.

As the target of this study was to identify the rumor tweets, the performances achieved by these classifiers still needed further improvements, as we aimed to achieve more accurate classification, especially with respect to the recall measure. Thus, the next section presents the ability of the ensemble classifiers in enhancing the overall recall and other measures when detecting the rumors.

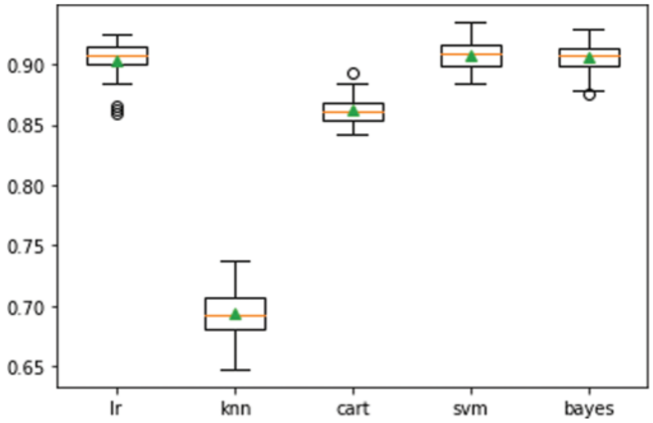

Figure 4: Boxplots of accuracy performance of the standalone classifiers

Table 7: Performance of Ensemble models with different base/weak classifier

Classifier name	Results							
	Accuracy		Recall		Precision		F-score	
	μ	σ	μ	σ	μ	σ	μ	σ
Random Forest	0.902	0.014	0.893	0.020	0.909	0.022	0.899	0.015
AdaBoost Classifier*	0.798	0.020	0.724	0.046	0.853	0.032	0.782	0.025
AdaBoost(SVM)	0.789	0.133	0.824	0.199	0.841	0.158	0.761	0.114
AdaBoost(LR)	0.857	0.022	0.805	0.040	0.900	0.026	0.849	0.025
Bagging Classifier*	0.869	0.016	0.801	0.024	0.931	0.017	0.860	0.020
Bagging(LR)	0.900	0.015	0.875	0.029	0.924	0.019	0.898	0.018
Bagging (KNN)	0.687	0.047	0.966	0.014	0.616	0.040	0.748	0.015
Bagging (CART)	0.868	0.017	0.799	0.024	0.929	0.018	0.854	0.018
Extra-Trees Classifier	0.904	0.014	0.910	0.024	0.903	0.024	0.906	0.015
Stacking Classifier (LR)	**0.917**	0.012	**0.987**	0.019	0.884	0.020	**0.933**	0.011
Stacking (SVM)	0.891	0.014	0.941	0.018	0.856	0.021	0.896	0.012
Stacking (NB)	0.886	0.015	0.920	0.024	0.862	0.024	0.891	0.014
Stacking (CART)	0.859	0.018	0.862	0.030	0.858	0.028	0.860	0.018
SGD Classifier	0.908	0.012	0.896	0.023	0.918	0.020	0.907	0.013

Note:
*Default settings.

4.2 Results of Model 2-Ensemble-based Machine Learning Models

We employed six ensemble classifiers (i) RF, (ii) AdaBoost, (iii) Bagging, (iv) ExtraTree, (v) Stacking-based ensemble classifiers, and (vi) SGD. Since the standalone classifiers presented in the previous section showed good performance, they can be used as base/weak classifier for the ensemble models. Tab. 7 shows the results of different ensemble-based models with different base classifiers. The stacking-based classifier with LR (Stacking-LR) base gives the highest performance in terms of accuracy 91.7%, recall, 0.987, and F-score, 0.933. It shows that the ensemble model also outperforms the standalone classifiers presented in Tab. 6. The results also show that the stacking-LR classifier achieves a robust performance using all repeated folds, as shown in Figs. 5–8.

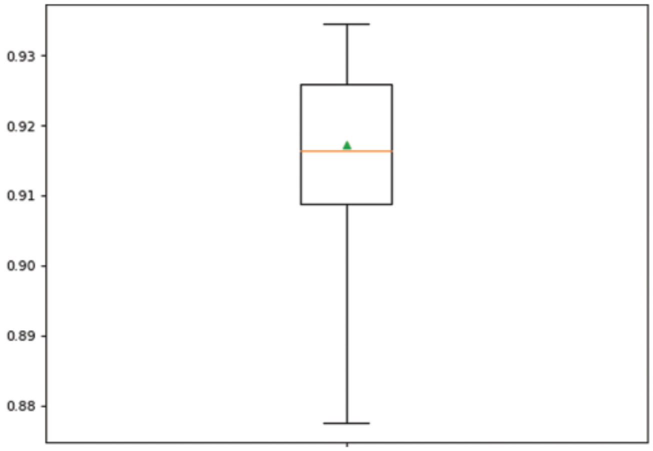

Figure 5: Boxplot of accuracy performance of Stacking-LR ensemble classifier over 10-fold repetition

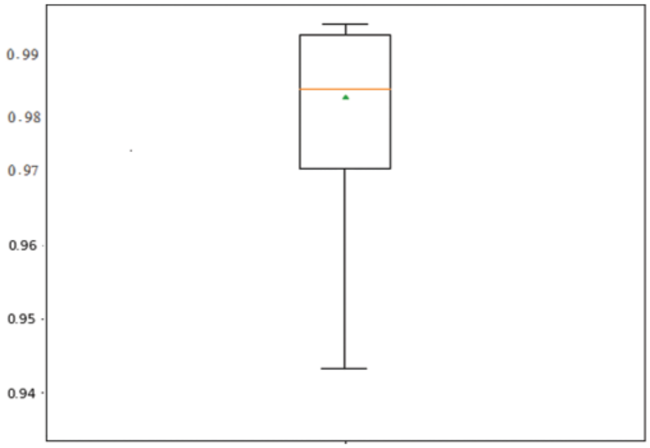

Figure 6: Boxplot of recall performance of Stacking-LR ensemble classifier over 10-fold repetition

4.3 Results of Model 3-Genetic Algorithm-based Support Vector Machine Model

As stated earlier, the extracted user-based and tweet-based features are fed into machine learning classifiers. Here, we conducted extensive experiments to select the best classifier that gives the highest performance. The genetic algorithm was used as tuning method for the hyper-parameters of each classifier. The TPOT[5] Python library was used for this purpose. The GA-based SVM gave the highest performance, with accuracy of 67.45%, as shown in Tab. 8.

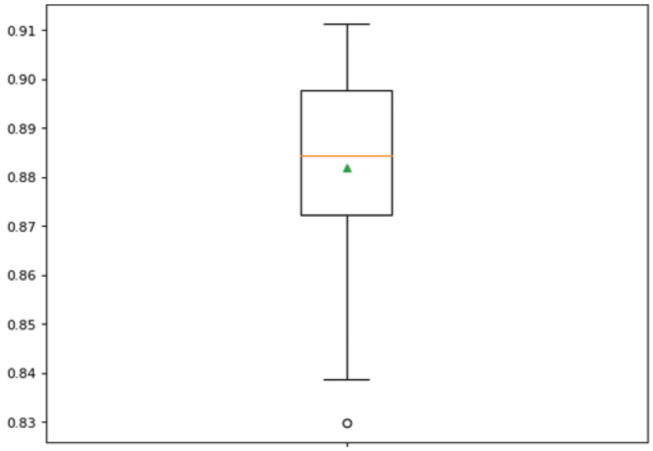

Figure 7: Boxplot of precision performance of Stacking-LR ensemble classifier over 10-fold repetition

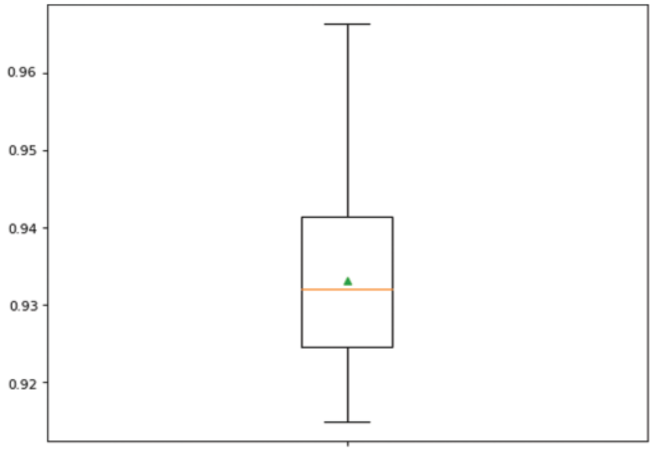

Figure 8: Boxplot of F-score performance of Stacking-LR ensemble classifier over 10-fold repetition

4.4 Results of the Proposed Model 4-Combined Genetic Algorithm-based Machine Learning Models with Stacking Ensemble

The proposed model combines the feature maps obtained by the second and third models (Stacking Classifier (LR) and GA-SVM). The classification results presented in Tab. 9 show the overall performance of the proposed model.

[5] http://epistasislab.github.io/tpot/

Table 8: Performance of ML classifiers with GA tuning

Classifier Type	Classifier name	Results			
		Accuracy	Recall	Precision	F-score
Standalone classifier	Linear Regression	0.645	0.621	0.566	0.557
	K-nearest Neighbor	0.623	0.642	0.598	0.560
	Decision Tree (CART)	0.657	0.645	0.639	0.560
	Support Vector Machine	**0.675**	0.529	**0.677**	**0.644**
	Naïve Bayes (Bernoulli)	0.663	**0.691**	0.672	**0.644**
	Random Forest	0.642	0.620	0.650	0.643
	AdaBoost	0.642	0.648	0.615	0.591
	Bagging	0.614	0.623	0.636	0.622
	ExtraTree	0.620	0.619	0.627	0.596

Table 9: Overall performance of the proposed model

	Accuracy	Recall	Precision	F-score
Non-rumor	92.63%	0.94	0.93	0.93
Rumor		0.91	0.92	0.94

As a summary, Tab. 10 shows the performance of all the applied models (the standalone machine learning model (SVM), ensemble machine learning model (stacking) and GA-SVM) compared with the proposed model. The results show that the proposed model outperforms all other models.

Table 10: Summarized classifier performance

	Accuracy (%)	Recall	Precision	F-score
Model 1 (SVM)	90.70	0.892	0.921	0.906
Model 2 (Stacking Ensemble)	91.70	**0.987**	0.884	0.933
Model 3 (GA-SVM)	67.45	0.530	0.677	0.644
The Proposed Model	**92.63**	0.930	**0.925**	**0.935**

5 Performance of the Similarity Techniques in the Tracking Phase

In order to track the source of the rumors, several similarity techniques were used in order to find the most similar tweets (top 1%) to a given target rumor (query). In this study, Cosine-based similarity, Jaccard-based similarity and Chebyshev distance (with Glove word embedding) were used. Fig. 9 shows the similarity score between the first rumor tweet in the dataset and the remaining tweets detected at the previous stage. The Chebyshev distance gives better insight into the similar tweets.

In order to evaluate the performance of the applied similarity techniques, 10 target rumors (queries) were chosen randomly from the tweets that were classified as rumors in the previous stage. For each query, different similarity techniques were then applied to compute the similarity between this query and all tweets that were classified as rumors in the detection phase (about 1371 out of 1480 tweets in the current dataset). The top 1% of similar tweets were then selected for this query and each similarity measure. After that, the ROUGE L values (precision, recall and F-measure) were calculated between this query and the obtained tweets in the top 1% list of each similarity measure. The precision, recall and F-measure values were then averaged for each similarity measure.

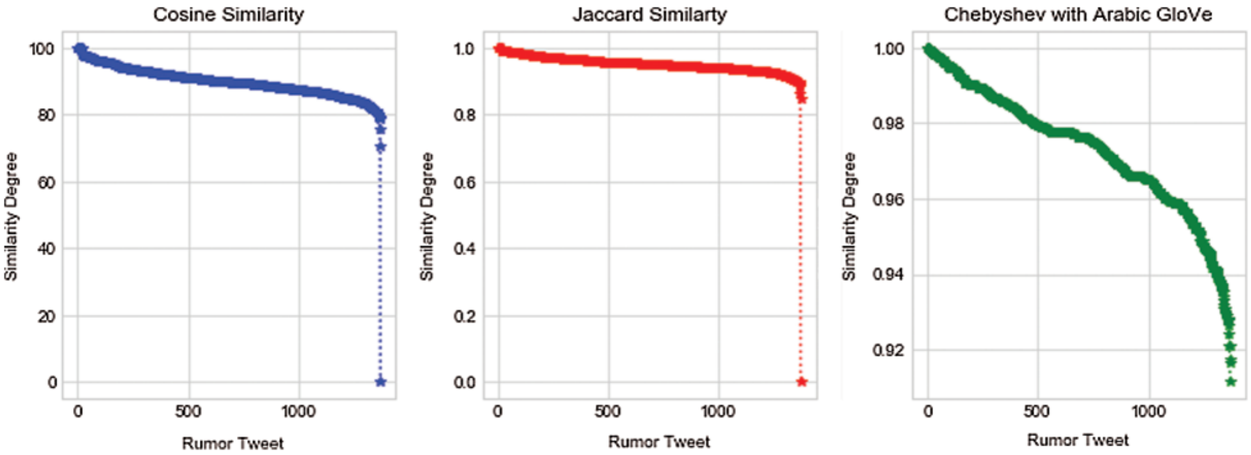

Figure 9: Similarity scores between Tweet ID:12 and remaining tweets

For instance, the top 1% of similar tweets for the first query (with ID 12) using Jaccard-based similarity were the tweets with IDs: 1479, 504, 498, 507, 487, 496, 495, 494, 490, 489, 488, 107, 62 and 12. The ROUGE L (precision, recall and F-measure) values were computed between this query and these obtained tweets in order to evaluate the performance of Jaccard-based similarity. The ROUGE L values are shown in Tab. 11.

Tabs. 12–14 show the summary of the ROUGE L (precision, recall and F-measure) values using Jaccard, Cosine and Chebyshev similarity techniques and the 10 used queries. The results show that the Chebyshev similarity technique obtained the best average ROUGE L (precision, recall and F-score) values using all queries compared to the Jaccard and Cosine similarity measures.

Thus, in order to detect and track the source of rumor, the target tweet/post will be examined using the proposed model (Combined Stacking Classifier (LR) and GA-SVM) to check whether this tweet is classified as rumor or not. The model proposed in this study is recommended to be used because it obtained the best performance compared to other rumor detection methods.

If the tweet/post is classified as a rumor, then the Chebyshev similarity technique will be used to compute the similarity between this tweet and all previously classified rumors. The top 1% of similar tweets will be obtained and the details of these tweets, such as creation date (created at), will help to recognize the source of the target tweet (rumor).

Table 11: ROUGE measure of query ID12

Query	Top 1% similar tweets	Precision	Recall	F-Measure
12	12	1	1	1
12	107	0.915	0.827	0.869
12	62	0.907	0.942	0.925
12	488	0.455	0.086	0.145
12	507	0.385	0.086	0.141
12	487	0.357	0.086	0.139
12	496	0.313	0.086	0.135
12	490	0.261	0.103	0.148
12	504	0.250	0.034	0.061
12	498	0.222	0.138	0.171
12	489	0.208	0.086	0.122
12	1479	0.164	0.155	0.159
12	494	0.150	0.103	0.122
12	495	0.085	0.069	0.076
Average		**0.405**	**0.272**	**0.301**

Table 12: Jaccard-based similarity with Glove word embedding

Query	Precision	Recall	F-score
12	0.383	0.257	0.285
23	0.149	0.179	0.153
145	0.194	0.203	0.193
67	0.463	0.461	0.461
1346	0.199	0.174	0.182
349	**0.852**	**0.734**	**0.786**
234	0.334	0.291	0.303
1400	0.384	0.308	0.336
783	0.171	0.220	0.186
981	0.231	0.160	0.177
Average	**0.336**	**0.299**	**0.306**

Table 13: Cosine-based similarity with Glove word embedding

Query	Precision	Recall	F-score
12	0.168	0.084	0.108
23	0.078	0.112	0.082
145	0.062	0.075	0.066
67	0.061	0.042	0.048
1346	0.131	0.125	0.122
349	0.230	0.177	0.190
234	0.145	0.155	0.146
1400	0.122	0.093	0.100
783	0.162	0.180	0.158
981	0.162	0.131	0.138
Average	**0.132**	**0.117**	**0.116**

Table 14: Chebyshev -based similarity with Glove word embedding

Query	Precision	Recall	F-score
12	0.269	0.147	0.171
23	0.147	0.176	0.152
145	0.231	0.224	0.226
67	0.532	0.520	0.523
1346	0.218	0.217	0.215
349	0.875	0.795	0.830
234	0.339	0.282	0.300
1400	0.391	0.323	0.349
783	0.221	0.279	0.236
981	0.196	0.174	0.180
Average	**0.342**	**0.314**	**0.318**

Since Twitter APIs provide us with the ability to obtain the time-stamp of each tweet and its retweets, it is easy to track the temporal diffusion of a rumor on Twitter. Fig. 10 presents an illustration of diffusion of rumor tweets over the time, where the bird represents the original tweet. and the arrows represent retweets. The *x*-axis represents time. Since the number of rumor tweets and retweets in real time can be huge, the algorithm that is presented in Section 3 suggests first reducing the number of examined tweets by eliminating the retweets from the search space, since the algorithm gives all credit to the original tweet, no matter who retweeted to whom. In addition, the algorithm reorders the tweets and their k top similar tweets according to the time when the tweet was posted. In case of two or more tweets shared the same time, we consider these tweets and the users who posted them as candidate rumor sources.

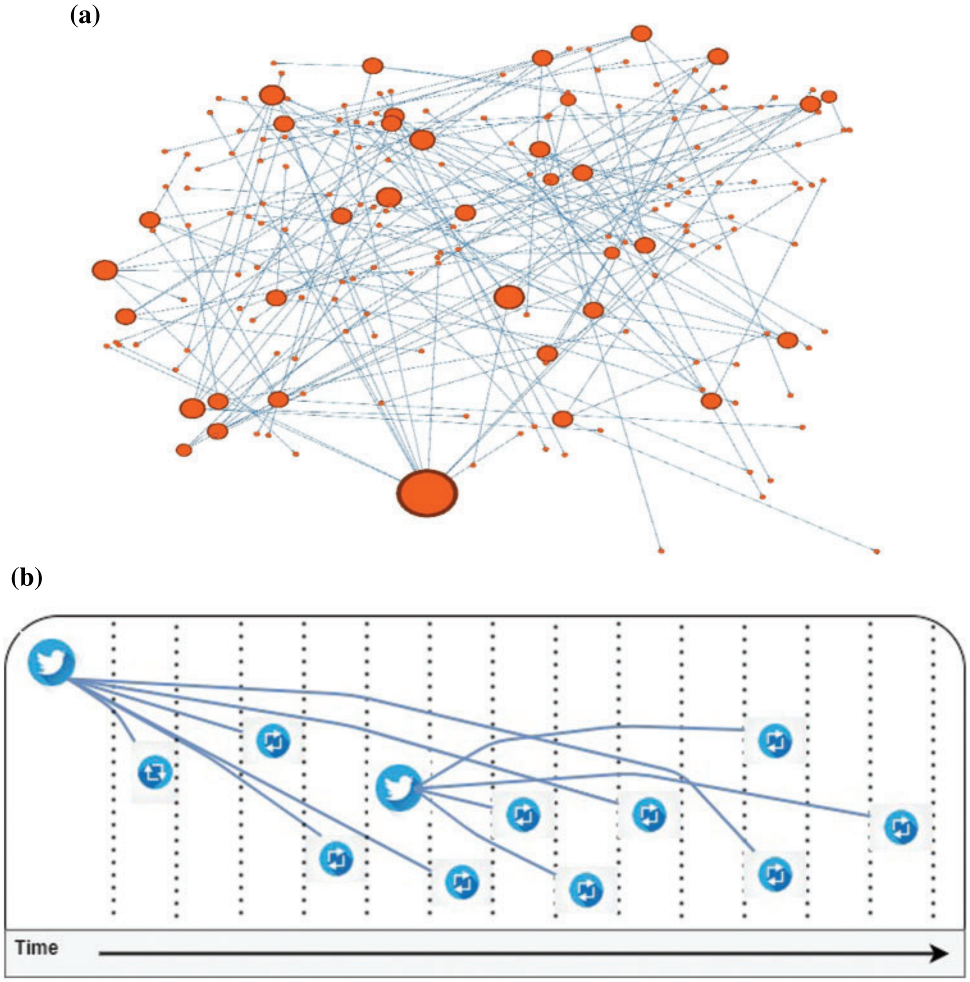

Figure 10: Diffusion of rumors over time: a) the search space b) an illustrative example of how the Algorithm 2 works

6 Conclusions and Future Research

In this study, the issues of detecting and tracking the source of rumors were investigated for COVID 19-related data for enhancing the stability of healthcare. A comprehensive approach was proposed that includes two phases: rumor detecting and tracking. In the first phase, several standalone and ensemble machine learning methods, including Linear Regression, K-nearest Neighbor, Decision Tree (CART), Support Vector Machine, Naïve Bayes (Bernoulli), Random Forest, AdaBoost, Bagging, Extra-Trees and Stacking were used. A new model was then proposed by combining two models, which are the Stacking Classifier (LR) and GA-SVM. The experimental results showed that the best standalone machine learning method was SVM, which obtained the best accuracy and F1-Score (0.907 and 0.906 respectively) comparing to other standalone machine learning methods. In order to improve the detection performance, ensemble learning methods were used, and the results showed that the Stacking Classifier (LR) could improve the performance in detecting rumors. The obtained accuracy, recall and F1-Score for the Stacking Classifier (LR) were 0.917, 0.987 and 0.933 respectively, which are the best findings compared to other standalone and ensemble machine learning methods. The proposed model was then applied, which achieved 0.926, 0.930 and 0.935 for accuracy, recall and F1-Score, which outperformed the other models.

For the second phase, several similarity techniques were used which were Cosine-based similarity, Jaccard-based similarity and Chebyshev (with Glove word embedding). The ROUGE evaluation measure was used to evaluated the effectiveness of these similarity techniques by applying 10 queries and obtaining the top 1% of similar tweets for each query, using each similarity technique. The ROUGE L (precision, recall and F-score) values obtained by applying Chebyshev-based similarity were the best (0.34, 0.31 and 0.32 respectively). Therefore, this study recommends applying the proposed model (combined Stacking Classifier (LR and GA-SVM) in the rumor detection phase and the Chebyshev-based similarity technique for the rumor tracking phase for COVID 19 related rumors that are posted in the Arabic language. Future work can examine the performance of different standalone and ensemble classifiers with different hyper-parameter tuning methods. In addition, more tweets can be collected to enrich the dataset used, in order to train the models on larger datasets.

Acknowledgement: The authors would like to thank Deanship of Scientific Research at Al Imam Mohammad ibn Saud Islamic university, Saudi Arabia, for financing this project under the grant no. (20-12-18-013).

Funding Statement: This research was funded by the Deanship of Scientific Research, Imam Mohammad Ibn Saud Islamic University, Saudi Arabia, Grant No. (20-12-18-013).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. M. Alzanin and A. M. Azmi, "Detecting rumors in social media: A survey," *Procedia Computer Science*, vol. 142, pp. 294–300, 2018.

[2] F. Liu, A. Burton-Jones and D. Xu, "Rumors on social media in disasters: Extending transmission to retransmission," in *Proc. of Pacific Asia Conf. on Information Systems*, Chengdu, China, pp. 49, 2014.

[3] L. Wu, J. Li, X. Hu and H. Liu, "Gleaning wisdom from the past: early detection of emerging rumors in social media," in *Proc. of the 2017 SIAM Int. Conf. on Data Mining, Society for Industrial and Applied Mathematics*, Houston, Texas, USA, pp. 99–107, 2017.

[4] L. Zhu, M. Liu and Y. Li, "The dynamics analysis of a rumor propagation model in online social networks," *Physica A: Statistical Mechanics and its Applications*, vol. 5, no. 20, pp. 118–137, 2019.

[5] M. Zhai, "The generation mechanism of internet rumors-based on consideration of information philosophy," *Multidisciplinary Digital Publishing Institute Proc.*, vol. 1, no. 3, pp. 114, 2017.

[6] M. Li, H. Zhang, P. Georgescu and T. Li, "The stochastic evolution of a rumor spreading model with two distinct spread inhibiting and attitude adjusting mechanisms in a homogeneous social network," *Physica A: Statistical Mechanics and its Applications*, vol. 562, pp. 125321, 2021.

[7] H. Yin, Z. Wang and Y. Gou, "Rumor diffusion and control based on double-layer dynamic evolution model," *IEEE Access*, vol. 8, pp. 115273–115286, 2020.

[8] C. Zhao, L. Li, H. Sun and H. Yang, "Multi-scenario evolutionary game of rumor-affected enterprises under demand disruption," *Sustainability*, vol. 13, no. 1, pp. 360, 2021.

[9] S. Kim and S. Kim, "Impact of the Fukushima nuclear accident on belief in rumors: The role of risk perception and communication," *Sustainability*, vol. 9, no. 12, pp. 2188, 2017.

[10] S. Kim and S. Kim, "Analysis of the impact of health beliefs and resource factors on preventive behaviors against the COVID-19 Pandemic," *International Journal of Environmental Research and Public Health*, vol. 17, no. 22, pp. 8666, 2020.

[11] S. Kim and S. Kim, “The crisis of public health and infodemic: Analyzing belief structure of fake news about COVID-19 pandemic,” *Sustainability*, vol. 12, no. 23, pp. 9904, 2020.

[12] L. Zhang, K. Chen, H. Jiang and J. Zhao, “How the health rumor misleads people’s perception in a public health emergency: Lessons from a purchase craze during the covid-19 outbreak in China,” *International Journal of Environmental Research and Public Health*, vol. 17, no. 19, pp. 7213, 2020.

[13] S. A. Alkhodair, S. H. Ding, B. C. Fung and J. Liu, “Detecting breaking news rumors of emerging topics in social media,” *Information Processing & Management*, vol. 57, no. 2, pp. 102018, 2020.

[14] M. Al-Sarem, W. Boulila, M. Al-Harby, J. Qadir and A. Alsaeedi, “Deep learning-based rumor detection on microblogging platforms: A systematic review,” *IEEE Access*, vol. 7, pp. 152788–152812, 2019.

[15] T. Chen, X. Li., H. Yin and J. Zhang, “Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection,” In: M. Ganji, L. Rashidi, B. C. M. Fung, C. Wang. (Eds.), *Trends and Applications in Knowledge Discovery and Data Mining*, pp. 40–52. Cham: Springer International Publishing, 2018.

[16] L. Wu, Y. Rao, H. Yu, Y. Wang and A. Nazir, “False information detection on social media via a hybrid deep model,” In: S. Staab, O. Koltsova, D. I. Ignatov. (Eds.), *Social Informatics*, Switzerland: LNCS, Springer Nature, pp. 323–333, 2018.

[17] A. Roy, K. Basak, A. Ekbal and P. Bhattacharyya, “A deep ensemble framework for fake news detection and classification,” arXiv Preprint ArXiv:1811.04670, 2018.

[18] S. Shelke and V. Attar, “Source detection of rumor in social network: A review,” *Online Social Networks and Media*, vol. 9, pp. 30–42, 2019.

[19] C. Shao, G. L. Ciampaglia, A. Flammini and F. H. Menczer, “A platform for tracking online misinformation,” in *Proc. of the 25th Int. Conf. Companion on World Wide Web (WWW ’16 Companion)—Int. World Wide Web Conf. Steering Committee*, Republic and Canton of Geneva, Switzerland, pp. 745–750, 2016.

[20] D. Shah and T. Zaman, “Rumors in a network: Who’s the culprit?,” *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

[21] Z. Wang, W. Dong, W. Zhang and C. W. Tan, “Rumor source detection with multiple observations: fundamental limits and algorithms,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 1, pp. 1–13, 2014.

[22] D. Shah and T. Zaman, “Rumor centrality: A universal source detector,” in *Proc. of the 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. on Measurement and Modeling of Computer Systems, SIGMETRICS ’12*, New York, NY, USA, pp. 199–210, 2012.

[23] P. D. Yu, C. W. Tan and H. L. Fu, “Rumor source detection in finite graphs with boundary effects by message-passing algorithms,” in *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, Cham, Springer, pp. 175–192, 2018.

[24] W. Xu and H. Chen, “Scalable rumor source detection under independent cascade model in online social networks,” in *11th IEEE Int. Conf. on Mobile Ad-hoc and Sensor Networks*, Shenzhen, China, pp. 236–242, 2015.

[25] J. Jiang, S. Wen, S. Yu, Y. Xiang and W. Zhou, “Rumor source identification in social networks with time-varying topology,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 166–179, 2016.

[26] J. Choi, S. Moon, J. Woo, K. Son, J. Shin *et al.,* “Rumor source detection under querying with untruthful answers,” in *IEEE Conf. on Computer Communications*, Atlanta, GA, USA, pp. 1–9, 2017.

[27] J. Choi, S. Moon, J. Shin and Y. Yi, “Estimating the rumor source with anti-rumor in social networks,” in *24th IEEE Int. Conf. on Network Protocols*, Singapore, pp. 1–6, 2016.

[28] D. T. Nguyen, N. P. Nguyen and M. T. Thai, “Sources of misinformation in online social networks: Who to suspect?,” in *IEEE Military Communications Conf.*, Orlando, FL, USA, pp. 1–6, 2012.

[29] W. Zang, P. Zhang, C. Zhou and L. Guo, “Discovering multiple diffusion source nodes in social networks,” *Procedia Computer Science*, vol. 29, pp. 443–452, 2014.

[30] H. T. Nguyen, P. Ghosh, M. L. Mayo and T. N. Dinh, "Multiple infection sources identification with provable guarantees," in *Proc. of the 25th ACM Int. on Conf. on Information and Knowledge Management (CIKM'16)*, New York, NY, USA, Association for Computing Machinery, pp. 1663–1672, 2016.

[31] A. Ghenai, "Health misinformation in search and social media," in *Proc. of the 2017 Int. Conf. on Digital Health (DH'17)*, London, United Kingdom, pp. 235–236, 2017.

[32] A. Ghenai and Y. Mejova, "Catching Zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on twitter," ArXiv Preprint ArXiv:1707.03778, 2017.

[33] T. Mondal, P. Pramanik, I. Bhattacharya, N. Boral and S. Ghosh, "Analysis and early detection of rumors in a post disaster scenario," *Information Systems Frontiers*, vol. 20, no. 5, pp. 961–979, 2018.

[34] F. Saeed, W. M. Yafooz, M. Al-Sarem and E. A. Hezzam, "Detecting health-related rumors on twitter using machine learning methods," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, pp. 324–332, 2020.

[35] S. Tasnim, M. M. Hossain and H. Mazumder, "Impact of rumors and misinformation on COVID-19 in social media," *Journal of Preventive Medicine and Public Health*, vol. 53, no. 3, pp. 171–174, 2020.

[36] F. Haouari, M. Hasanain, R. Suwaileh and T. Elsayed, "ArCOV19-rumors: Arabic COVID-19 twitter dataset for misinformation detection," ArXiv Preprint ArXiv:2010.08768, 2020.

[37] C. Li, F. Liu and P. Li, "Text similarity computation model for identifying rumor based on Bayesian network in microblog," *The International Arab Journal of Information Technology*, vol. 17, no. 5, pp. 731–741, 2020.

[38] R. Sicilia, S. L. Giudice, Y. L. Pei, M. Pechenizkiy and P. Soda, "Twitter rumour detection in the health domain," *Expert Systems with Applications*, vol. 110, pp. 33–40, 2018.

[39] D. K. Vishwakarma, D. Varshney and A. Yadav, "Detection and veracity analysis of fake news via scrapping and authenticating the web search," *Cognitive Systems Research*, vol. 58, pp. 217–229, 2019.

[40] R. K. Kaliyar, A. Goswami, P. Narang and S. Sinha, "FNDNet—A deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, pp. 32–44, 2020.

[41] Y. Zhang, W. Chen, C. K. Yeo, C. T. Lau and B. S. Lee, "Detecting rumors on online social networks using multi-layer auto-encoder," in *2017 IEEE Technology and Engineering Management Conf.*, Santa Clara, CA, pp. 437–441, 2017.

[42] A. S. Torshizi and A. Ghazikhani, "Automatic twitter rumor detection based on LSTM classifier," In: L. Grandinetti, S. Mirtaheri, R. Shahbazian. (Eds.), *High-Performance Computing and Big Data Analysis—TopHPC 2019: Communications in Computer and Information Science*, vol. 891. Cham: Springer, 2019.

[43] O. Ajao, D. Bhowmik and S. Zargari, "Fake news identification on twitter with hybrid CNN and RNN models," in *Proc. of the 9th Int. Conf. on Social Media and Society (SMSociety'18)*, New York, NY, USA, Association for Computing Machinery, pp. 226–230, 2018.

[44] A. Albahr and M. Albahar, "An empirical comparison of fake news detection using different machine learning algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, pp. 146–152, 2020.

[45] Z. Wu, D. Pi, J. Chen, M. Xie and J. Cao, "Rumor detection based on propagation graph neural network with attention mechanism," *Expert Systems with Applications*, vol. 158, pp. 113595, 2020.

[46] H. Zhang, M. A. Alim, X. Li, M. T. Thai and H. T. Nguyen, "Misinformation in online social networks: Detect them all with a limited budget," *ACM Transaction on Information System*, vol. 34, no. 3, pp. 1–24, 2016.

[47] Z. Zhao, P. Resnick and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proc. of the Twenty-forth Int. Conf. World Wide Web*, Florence, Italy, pp. 1395–1405, 2015.

[48] S. Vosoughi, N. Mohsenvand and D. K. Roy, "Rumor gauge: Predicting the veracity of rumors on twitter," *ACM Transaction on Knowledge Discovery from Data*, vol. 11, no. 4, pp. 1–38, 2017.

[49] Z. Wang, W. Dong, W. Zhang and C. W. Tan, "Rooting our rumor sources in online social networks: The value of diversity from multiple observations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 663–677, 2015.

[50] A. Louni and K. P. Subbalakshmi, “Who spread that rumor: Finding the source of information in large online social networks with probabilistically varying internode relationship strengths,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 335–343, 2018.

[51] A. Maryam and R. Ali, “Misinformation source identification in an online social network,” in *IEEE 5th Int. Conf. for Convergence in Technology (I2CT)*, Bombay, India, pp. 1–5, 2019.

[52] F. Ji, W. P. Tay and L. R. Varshney, “An algorithmic framework for estimating rumor sources with different start times,” *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2517–2530, 2017.

[53] N. Antulov-Fantulin, A. Lancic, T. Smuc, H. Stefancic and M. Sikic, “Identification of patient zero in static and temporal networks: Robustness and limitations,” *Physical Review Letters*, vol. 114, no. 24, pp. 248701, 2015.

[54] D. Shah and T. Zaman, “Detecting sources of computer viruses in networks: theory and experiment,” in *Proc. of the ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Computer Systems*, New York, USA, Columbia University, pp. 203–214, 2010.

[55] L. Shu, M. Mukherjee, X. Xu, K. Wang and X. Wu, “A survey on gas leakage source detection and boundary tracking with wireless sensor networks,” *IEEE Access*, vol. 4, pp. 1700–1715, 2016.

[56] C. H. Comin and L. Da Fontoura Costa, “Identifying the starting point of a spreading process in complex networks,” *Physical Review E*, vol. 84, no. 4, pp. 56105, 2011.

[57] J. Jiang, S. Wen, S. Yu, Y. Xiang and W. Zhou, “Identifying propagation sources in networks: State-of-the-art and comparative studies,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 465–481, 2017.

[58] T. Eldeeb, “GloVe model for distributed arabic word representation,” 2018, [Online]. Available: https://github.com/tarekeldeeb/GloVe-Arabic. [Accessed on 10 November 2020].

[59] F. Haouari, M. Hasanain, R. Suwaileh, and T. Elsayed, “Arcov-19: The first arabic covid-19 twitter dataset with propagation networks,” in *Proc. of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Virtual), pp. 82–91, 2021.

[60] M. Al-Sarem, F. Saeed, A. Alsaeedi, W. Boulila and T. Al-Hadhrami, “Ensemble methods for instance-based Arabic language authorship attribution,” *IEEE Access*, vol. 8, pp. 17331–17345, 2020.

[61] Y. Wu, L. Liu, Z. Xie, J. Bae, K. H. Chow *et al.,* “Promoting high diversity ensemble learning with ensemble bench,” ArXiv Preprint ArXiv:2010.10623, 2020.

[62] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[63] R. E. Schapire, “A brief introduction to boosting,” in *IJCAI’99: Proc. of the 16th Int. Joint Conf. on Artificial Intelligence*, Stockholm, Sweden, vol. 2, pp. 1401–1406, 1999.

[64] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[65] T. Zhang, “Solving large scale linear prediction problems using stochastic gradient descent algorithms,” in *Proc. of the Twenty-First Int. Conf. on Machine Learning*, Banff, Alberta, Canada, pp. 116, 2004.

[66] S. Džeroski and B. Ženko, “Is combining classifiers with stacking better than selecting the best one?,” *Machine Learning*, vol. 54, no. 3, pp. 255–273, 2004.