Al-Ahgaff University Journal of Computer Science and Mathematics
مجلة جامعة الأحقاف لعلوم الحاسوب والرياضيات

Vol. 2, October 2024, pp. 1~22

Arabic Rumor Event-based Detection on Twitter using Attention LSTM Models and Text Features

Arwa AlAttas
Faculty of Graduate Studies, Al-Ahgaff University, Mukalla, Yemen
arwaahmedats@gmail.com
Hassen A. Mogaibel
Information Technology Department, Al-Ahgaff University, Mukalla, Yemen
Hassen.mogaibel@gmail.com
Mohammed Salem Binwahlan
Faculty of Applied Sciences, Seiyun University, Seiyun, Yemen
Moham2007med@yahoo.com

Article Info

Article history:

Received June 20 2023
Accepted July 25, 2023

Keywords:

Rumors Detection
Deep Learning
Event level
Twitter
Arabic

ABSTRACT

When news propagates on Twitter, it is published, circulated, and discussed through a series of tweets. Rumor spreaders take advantage of the rapid spread to bring about harmful effects on society and economy while making recipients to bias their opinion and beliefs to the wrong side. Detecting rumors on social media at an early stage is critical to minimizing their catastrophic and tragic effects. Many researchers have sought effective deep learning-based solutions to detect rumors on Twitter, but they were limited to certain languages, such as English and Chinese. Few deep learning models have been used to detect Arabic rumors; however, those studies were either used to detect rumors on specific topics, losing the advantage of generalization, or were only concerned with detecting rumors at the tweet-level, ignoring event-level detection. This paper aimed to introduce two deep learning models to perform Event-level rumor detection on Twitter's Arabic content efficiently utilizing latent textual features extracted directly from tweets' texts. Two variations of Recurrent Neural Networks have been utilized: LSTM and Bi-LSTM. An attention mechanism was embedded in the models to optimize the performance of these models to achieve optimal results. Both proposed models achieved significant results where the LSTM and Bi-LSTM models reached at most 96.37% and 96.53% of accuracy respectively. Embedding the attention mechanism noticeably improved the performance of the models where LSTM-Attention and Bi-LSTM-Attention models reached accuracy of 96.41% and 96.61% respectively.

الخلاصة

عندما ينتشر الخبر على تويتر ، يتم تداوله ومناقشته من خلال سلسلة تغريدات. يستفيد ناشرو الشائعات من هذا الانتشار السريع لإحداث آثار ضارة على المجتمع والاقتصاد مع جعل المتلقين يتحيزون بآرائهم ومعتقداتهم إلى الجانب الخاطيء. يعد اكتشاف الشائعات على وسائل التواصل الاجتماعي في مراحل الانتشار المبكرة أمرًا بالغ الأهمية لتقليل آثارها الكارثية والمأساوية. سعى العديد من الباحثين إلى حلول فعالة قائمة على التعلم العميق للكشف عن الشائعات على تويتر ، لكنها اقتصرت على لغات معينة ، مثل الإنجليزية والصينية. تم استخدام عدد قليل من نماذج التعلم العميق للكشف عن الشائعات العربية. ومع ذلك ، فقد تم استخدام هذه الدراسات إما للكشف عن الشائعات حول موضوعات محددة مما أفقدها ميزة التعميم ، أو كانت خذة النماذج معنية فقط باكتشاف الشائعات على مستوى التغريدات ، وتجاهل الكشف على مستوى الحدث. هدفت هذه الورقة البحثية إلى تقديم نموذجين للتعلم العميق لأداء اكتشاف الشائعات على مستوى الأحداث على محتوى تويتر باللغة العربية بكفاءة باستخدام السمات النصية الكامنة المستخرجة مباشرة من نصوص التغريدات. تم استخدام نوعين مختلفين من الشبكات العصبية المتكررة ؛ LSTM و Bi-LSTM. تم تضمين آلية الانتباه في النماذج لتحسين أداء هذه النماذج لتحقيق أفضل النتائج. حقق كلا النموذجين المقترحين نتائج مهمة حيث بلغ النموذجان LSTM و , Bi-LSTM دقة 96.37٪ و 96.53٪ على التوالي. أدى تضمين آلية الانتباه إلى تحسين أداء النماذج بشكل ملحوظ حيث وصل نموذج LSTM-Attention و Bi-LSTM-Attention إلى دقة بلغت 96.41٪ و 96.61٪ على التوالي.

1. INTRODUCTION

As seen today, social networks reflect a real-life social interaction paradigm in which users are able to keep in touch with their friends and meet new people regardless of geographical, cultural, or time-related limits.[1]. The information on those networks has no guarantee on quality and credibility because any user registered in a social network can be a reporter or a source of information. The nature of social media also provides a rich and convenient environment for rumormongers to post and spread false stories and information, which can lead to major chaos and unpredictable reactions from those involved. Twitter, in particular, contributes to the rapid spread of rumors and unverified news due to its nature and the characteristics on which it was built. It is known as the Retweet feature, which allows any user to share the content of another user with all of his or her followers and their followers, thereby spreading a single post to thousands or even millions of users.

Twitter has recently become the top-one source of news transmission and exchanging; the platform is often targeted by celebrities and well-known personalities to share their news and opinions with their fans. It is important to note that when news spreads on Twitter, it would be published, transmitted, and discussed through a series of tweets, especially since tweet content is limited to 280 characters. Furthermore, the news can be published without any comprehensive or prior investigation or even checking the veracity of the news or the publisher's background. Rumor spreaders use Twitter's rapid spread to manipulate public opinion, causing harmful political and economic changes and biasing recipients' opinions and beliefs to the wrong side.

The rapid spread of false information and rumors usually causes negative effects negative effects and cause severe damage at the level of individuals and society, these negative effects may be represented in defaming and discredit individuals or the organizations and official bodies.

An example is the nuclear disaster of Fukushima Daiiachi that happened in Japan in 2011. At that time, a rumor widely spread on the China's online social network - Sina Weibo, saying that iodized salt can protect people from the radiation effects. This rumor increased the salt price by almost five to ten times, because people rushed to buy much more iodized salt than they need [2]. One of the real-life examples was mentioned by [2]:"on April 23rd 2013, a fake news claiming two explosions happened in the White House and Barack Obama got injured was posted by a hacked Twitter account named Associated Press. Although the White House and Associated Press assured the public minutes later the report was not true, the fast diffusion to millions of users had caused severe social panic, resulting in a loss of $136.5 billion in the stock market."

Rumor detection is the process of determining if a given piece of information is rumor or non-rumor. Detecting rumors on social networks is critical to minimizing their catastrophic and tragic effects on the social and economic sides of society, and these incidents of false information demonstrated the vulnerability of social media to rumors, as well as the need for effective methods to detect rumors at earlier stages of emerging.

In this paper, two deep learning models based on Recurrent Neural Network (RNN) for Arabic event rumor detection on Twitter would be introduced; the main contributions can be summarized as follow:

(1) Built deep neural network models for automatically detecting rumors on Twitter's Arabic content at the event-level utilizing text features only.

(2) Embedding an attention mechanism into the model to optimize the performance.

The remainder of this paper is organized as follows: Section 1 introduces the problem and overviews some of the recent studies in Arabic rumor detection. Then in section 2, the methodology that has been followed to create Arabic event-based rumor detection framework would be described. The experiments and results are discussed in section 3.

1.2 Definition of Rumours

A. **Rumour**

Rumours are circulating statements or stories with unverified veracity or deliberately false. DiFonzo and Bordia [3] described a rumour as, "unverified and instrumentally relevant information statements in circulation that arise in contexts of ambiguity, danger or potential threat, and that function to help people make sense and manage risk". Rumour is defined as well as a diffusive piece of information which has unverified or intentionally false truth value [4]. In social networks, a rumour is a claim made by social media users whose truth has not been verified yet and can potentially spread beyond their private network. This claim can be proven later true or false [5]. On Twitter, a Rumour would be considered a group of posts (tweets) sharing assertions of the same unverified statement to propagate it through the network in many cascades [6].

B. **Mis-information**

Treen, et al. [7] provided a prevalent definition that agreed upon by this research ; they stated that misinformation refers to information that is incorrect, inaccurate or misleading. Note that misleading information does not have to be incorrect; rather, it may have been mentioned or presented outside of its proper context. In their quest to define Misinformation in social media specifically, Wu, et al. [8] aimed to define the boundaries between misinformation and related concepts that may confuse, such

as fake news and rumours. Still, in particular, they focused on disinformation since it is the most similar or confusing term. At first, they introduce Misinformation as false or erroneous information that has been purposefully created and spread, intentionally or accidentally. The authors then stated that Misinformation and disinformation relate to incorrect information. The fundamental distinction between them is whether the information is intentionally designed to deceive, with disinformation usually referring to purposeful and intentional incidents and Misinformation usually referring to unintentional ones. This study also agreed with Wu, et al. [8] who use "misinformation" as an umbrella term for all misleading or inaccurate information transmitted on social media, such as rumour, which refers to widely spread unverified information that may later turn out to be either incorrect or correct, and fake news, which refers to a piece of false information spread in the form of news.

C. Fake News

News can be characterized as a report of recent, intriguing, and noteworthy events or events that have a substantial impact on people and can be viewed as outcomes of journalism, which is meant to deliver "independent, reliable, accurate, and comprehensive information." Since the "primary purpose of journalism is to provide citizens with the information they need to be free and self-governing," media must report the truth above all else.

Lazer, et al. [9] described fake news as false material resembling news media content in form but not in organizational method or intent. In this way, fake news outlets lack the news media's editorial rules and methods for guaranteeing information accuracy and trustworthiness. Wu, et al. [8] mentioned that m fake news is purposely disseminated falsehoods in the shape of news. Recent events demonstrate that false news may be used as propaganda and spread via news media and social media. Lazer, et al. [9] also confirmed that fake news overlaps with other information.

1.3 Rumour Detection Approaches

During the last few years, many studies contributed to finding novel solutions for detecting rumors. Those studies can be categorized either based on the classification method approaches or detection level approaches. According to the used classification method the studies are categorized as follow:

A. Machine Learning Approach The earlier studies introduced models for rumour detection that relied on machine learning algorithms; these learning algorithms incorporated a wide variety of features and formulated rumour detection into a binary classification task (Rumour and non-Rumour). The extracted features are a mixture of content and context [5]. Content features are extracted directly from the text, such as linguistics features. On the other hand, context features rely on surrounding information. Context features could be extracted from users' characteristics, social network propagation and reactions of other users to the news or post.

B. **Deep Learning Approach** More effective techniques arose to take advantage of deep learning artificial neural networks in rumour detection, showing significant and promising results for many research fields, including text mining and NLP. The advantage of this approach over machine learning is the ability to learn and represent the needed features automatically[5, 10].It has the ability of mitigating feature engineering by fully utilizing its expressive capacity for modelling the features of input data[11].

In terms of the detection level, there are two approaches of mechanisms for detecting rumors circulating in various social media channels, as identified and explained by Han [12] in her dissertation:

A. Message-level Rumour Detection

Any rumor detection mechanism that attempts to determine if a particular post in the dataset to be a rumour or not. The objective behind this approach is to find multiple separate sub-events (i.e., rumors) that are associated with an event. The event in this situation is neutral and is not an indication of a rumour or a non-rumour. Message-level detection models accept any sort of message related to an event that has the potential to generate several rumors. An example of such events is the COVID-19 which was a global trending event in which social media platforms were used to disseminate different facts, warnings, general and protective information, statistics, as well as news and rumors. In this research, this type of detection referred to as **Tweet-level Detection**.

B. **Event-Level Rumour Detection**

Include any rumor detection mechanism that attempts to determine whether an event is associated with a rumour or not based on a collection of tweets associated with it. The primary distinction between this and message-level detection is that an event in the former is a specific rumour.

1.4 Related Works

The methods and techniques that have been utilized for Arabic rumour detection on twitter content can be categorized in two approaches according to the used methods. The primary approach is machine learning, which was adopted initially until a few researchers shifted toward employing the deep learning approach to develop effective methods for detecting Arabic rumors or employing it in related tasks such as detecting Fake News or Misinformation.

The Authors of [13] proposed one of the first studies focused on detecting Arabic false information on Twitter. They seek to provide a system that automatically evaluates the credibility of Arabic web material, with a focus on the news sector. Their method was able to determine whether an Arabic post was credible, not credible, or questionable.

For credibility assessment of Arabic Twitter news posts, the authors of [14] used a hybrid feature set model that combines topic-related and source-related features. According to the authors, the bulk of previous studies have focused on either topic-related or source-related features. While a variety of studies were concerned with Arabic, only a few of them used hybrid features to assess the credibility of Arabic material on Twitter. Three alternative classification algorithms which are: Decision Tree, support vector machine (SVM) and Naive Bayesian (NB).

One of the most recent research that addressed the problem of identifying Arabic rumors on Twitter was introduced by [15]. They employed an expectation-maximization algorithm (E-M) in two distinct learning schemas: semi-supervised learning and unsupervised learning. In order to achieve the best model performance, they planned to assess previously utilized features and select the top twelve topic-based features that obtained the best results with the E-M algorithm.

The authors of [16] presented a study centered with analyzing fake news, but their main goal was to introduce a novel corpus including rumors-related subjects. This study found that most prior researchers focused their efforts on detecting rumors after the rumors had propagate, indicating that there is a significant gap between rumor diffusion and rumor detection. Three machine learning classifiers have been employed for assessing the new dataset: Support Vector Machine (SVM), Decision Tree (DT), and Multinomial Naive Bayes (MNB). It has been noticed that fake news and rumors were considered the same thing in this study.

The framework presented by [17] designed for assessing the credibility of Arabic postings on Twitter. Multiple machine learning classifiers were investigated in this work, and after some experimentation, only four were preserved. Random forest, adaboost, and logistic regression were used as classifiers. In addition to content-based and user-based features, this study investigated the impact of sentiment analysis on producing additional features that improve the detection of fake news. The use of sentiment analysis has been shown to improve detection accuracy.

In the paper [18] The authors introduced the CAT credibility assessment model in attempt to develop a binary classification that labels a Twitter tweet as credible or not credible. Their model was trained using their own collected dataset and two classifiers: Random Forest and Decision Tree. To figure out whether only a single type of features may be employed as a credibility determinant, they conducted a comparison analysis of content-based features versus user-based features. However, the best results were obtained by combining both types of features, while user-based features exceed content-based features performance by only 0.02%.

In the paper[19], The authors proposed a machine learning approach for classifying tweets as credible or not credible. The authors had three main goals to focus on when developing this model: the first was proving that N-gram word-based features outperform content-based and source-based features. The second goal is to compare the model classifiers' performance on two distinct datasets: English and Arabic. Their final goal was to create a smartphone application capable of detecting the credibility of real-time datasets obtained from Twitter. They trained five machine learning classifiers: Nave Bayes (NB), Random Forest (RF), Linear Support Vector Machines (LSVM), K-Nearest Neighbor (KNN), and Logistic Regression (LR).

The study by [20] This was one of a number of studies that explored the COVID-19 Twitter content. This research helped to introduce one of the largest public Arabic COVID-19 databases. To validate and measure the quality of their dataset, they used two classification approaches: the first is Machine Learning, in which they built an SVM model, and the second is Deep Learning, in which they used AraBERT, a transformer-based model trained on Arabic news.

Paper[21] conducted an additional study that attempted to build a large COVID-19 Arabic dataset. The dataset was obtained from Arabic tweets and has been devoted for misinformation classification tasks. As a second contribution, the researchers created and trained two Word Embedding models while employing the whole gathered dataset, which contains over two million tweets. These two models were created using two word embedding techniques: word2vec and FASTTEXT. Their final contribution was to use a collection of Machine Learning and Deep learning techniques to detect misinformation. Support vector machine (SVM), multinomial naive Bayes (NB), Extreme Gradient Boosting (XGBoost), Random forest (RF), and Stochastic Gradient Descent (SGD) were the Machine Learning techniques that have been used. Convolutional neural networks (CNN), Recurrent Neural Networks with bidirectional long-short-term memory (RNN BiLSTM), and Convolutional Recurrent Neural Networks (CRNN) are among the techniques employed in deep learning.
Authors of [22] built the first Arabic dataset sourced from Twitter that is able to be utilized in misinformation detection tasks at both the tweet and event levels. Although the dataset was assembled for event-level classification tasks, the authors merely carried out tweet-level binary classification utilizing the bidirectional Graph Convolutional Networks model (Bi-GCN), PPC-RNN+CNN, AraBERT, and MARBERT.

The goal of [23] was to find out if the deep learning model could detect false news in Arabic COVID-19 tweets. They conducted a comparison analysis using three fundamental deep learning models: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Gated Recurrent Unit (GRU). Multiple transformer-based models, including AraBERT v1, AraBERT v02, AraBERT v2, ArElectra, QARiB, Arbert, and Marbert, were also utilized in their comparison. The researchers used two available datasets derived from Arabic Twitter content during the Corona pandemic: ArCOV19-Rumors and COVID-19-Fakes.

To detect Arabic rumors on Twitter, [24] presented a hybrid model of a Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). As stated, their contributions can be split into four objectives: first, to review existing research on rumor detection on social media; second, to use deep learning methods to classify text as rumor or non-rumor; third, to examine the use of word embedding over the traditional feature representation techniques and compare results; and last, to compare the efficiency of their model with other baseline methods. It should be emphasized that they used the same data collected by [15] and that the proposed framework is limited to detecting rumors based on the tweet level.

As stated by [25] they introduced an effective approach for detecting Arabic rumor tweets utilizing the eXtreme gradient boosting (XGBoost) algorithm. This algorithm has been trained with a set of topic-based features that have been generated from the content-based and user-based features. The authors follow the same proposed framework as [15] and used the same dataset published by them; apparently, they nearly replicated their work. Still, the difference is that they achieved supervised learning utilizing a different machine learning algorithm and employed all of [15]'s proposed topic-based features. In contrast, the latter used only the best 12 of their proposed features.

1.5 Problem Formulation

Detecting rumors on social networks at an early stage is critical for mitigating their detrimental impact. Many studies have sought efficient methods to identify rumors on Twitter. These studies belong to one of two categories: Machine Learning studies [13-19, 26-28] or Deep Learning studies [2, 4, 10, 20-22, 29-34].

As evidenced by the preceding section, the majority of the related studies employed a machine learning techniques to develop models that produced good results. However, The Machine Learning approach relies on manually crafted features that cannot automatically detect rumors during the earlier stages of diffusion and may affect the model's performance. Conversely, the deep learning techniques that have been employed to detect Arabic rumors on twitter content is relatively limited compared to certain languages, such as English and Chinese where they own a bigger share. Those studies related to the Arabic content in twitter suffer at least one of two limitations: First, some of the studies were dedicated to detecting rumors on specific topics of Twitter content, such as COVID-19, which loses the advantage of generalization. Second, most studies were concerned only with detecting rumors at the tweet level and neglecting event-level detection. However, it is much more logical and faster to detect rumors by defying the rumor as a whole and not each tweet separately, where some tweets will be ignored and will not be classified as a rumor.

The only studies that introduced a method to detect Arabic rumor events on Twitter utilizing machine learning with topic (event) based features were proposed by [15, 25]. The examined features in their studies are derived from the user and content features didn't include the linguistic features, which, according to Jain, et al. [35], help discriminate the misguiding aspects of the content utilized to hide the content's author's writing pattern. Furthermore, the handcrafted features cannot detect variations in the context of the posts [2].

This paper proposes a deep learning based framework for automatically detecting rumors of Twitter's Arabic content at the event level. Two variants of the Recurrent Neural Network would be employed, in which they would be fed with the textual features extracted from the events' tweets themselves. A soft attention mechanism would be embedded into the models to enable distinct feature extraction from high duplication and advanced importance focus that varies over time. This paper is an extension of a previous study by AlAttas, et al. [36],and the main objectives of the paper can be addressed as:

(1) To build a deep learning-based models that able to detect Arabic rumour on twitter at the Event-Level.
(2) To build a model that able to swiftly identify rumours in the early stages of their spread by relying on the latent textual features of tweets' text only.
(3) To evaluate and compare the efficiency of the proposed deep learning models.
(4) To evaluate and compare the efficiency of the proposed models before and after utilizing the soft attention mechanism.

2. METHODOLOGY

In this study a two deep learning models have been employed to analyze the Arabic content of twitter to detect rumors on event-level. One of the main objectives of this study is to address the problem of individually classifying a group of tweets related to one specific event or discussing the same topic as rumor or non-rumor. Toward achieving this objective, the dataset must be prepared probably where additional preprocessing steps must be taken in consideration. The proposed framework has four main phases that is shown in Figure 1. The four phases are data preprocessing, variable-length post series construction, Feature representation and classification model.

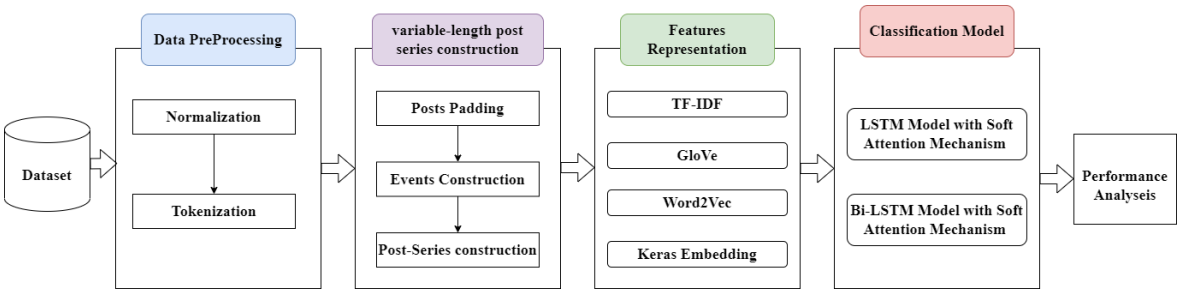


Figure 1 The Proposed Framework

2.1 Data Pre-Processing

There are major preprocessing steps performed to enhance the performance of the created cleaned Arabic tweeter dataset. Figure 3 shows the algorithm of the preprocessing module for each tweet in tweet dataset. It includes eight steps. These steps can be grouped into two categories namely: Normalizing and Tokenization.

Algorithm 1: Data Pre-Processing Algorithm

Input: Arabic twitter dataset
Output: Cleaned Dataset

```
Read the data from input file
Split input file into Event_id set, Class set and Tweets set
For each tweet in tweets
    Remove punctuation
    Remove diacritics
    Remove special characters
    Remove stop words
    Remove Latin characters
    Remove repeated letters
    Normalize Alef   //Unify different shapes of the letter أ
    Normalize Teh   //Unify different shapes of the letter ت
    Store the cleaned Tweets set along with Event_Id set and Class set in  cleaned
End for
End
```

A. **Normalization** is concerned with making more consistence dataset and it aim for linguistic reduction and standardization at the corpus level[37, 38]. In this procedure, the following steps have been relied upon[39]:

- Removing punctuation which is used to organize text and make it more readable.**Table1** contains the punctuations to be removed
- Removing Diacritics which are used to determine the letter's sound or pronunciation distinguish the Arabic language and play the same role as vowels in the English language. The diacritics to be removed are grouped in **Table1.**
- Removing stop words which refer to the common and most used words in a language. Stop words are categorized as adverbs, units of measurement, coin names, conditional pronouns, interrogative pronouns, prepositions, pronouns, reference names/determinants, relative pronouns, transformers (verbs, letters), and verbal pronouns. Stop words in the Arabic language are classified as words that can take suffixes or prefixes and words that can't take suffixes or prefixes. **Table 1** shows examples of Arabic stop-words
- Removing URLs which refers to Uniform Resource Locator; in other words webpages links or addresses. They usually started with “https//”
- Removing Latin characters also known as the Roman alphabet, is the collection of letters originally used by the ancient Romans to write the Latin and English language.
- Removing Elongation or repeated letters where sometimes users in social media tend to repeat some letters of the words, such as the word "شـكرا", which can be written as "شـكراا" The users do such things to increase their feeling.
- Removing non-Letter or special characters that comprise all the symbols except alphabet characters such as "&", "#", and "%".
- Unify letters with different shapes; some letters in Arabic have many shapes influenced by their position in the word and its diacritic and the surrounding letters' diacritics. An example is the letter "أ", which has the following shapes “ا ,إ, آ ,أ". Each letter from this list would be replaced with the corresponding general shape.

Arabic Punctuations	( )	{}	;	“	|	’	[ ]	:	؛
	!	,	?	‘	.				
Arabic Diacritics	ً	ٌ	ٍ	ِ	َ	ُ	ّ	ِ	ً
Arabic Stop Words	كلا	إن	إلى	من	على	فوق	تحت	لماذا	كيف
	إذن	إذا	عن						
Special Characters	#	=	$	~	&	%	×	_	-
	@	/	+	^	< >				

Table 1 Arabic Punctuations, diacritics, Stop words and Special Characters

Table 2 show examples of each step of the normalization procedure:

Data Sample 1	#هزه_ارضيه_المدينه_المنوره (وَمَا نُرْسِلُ بِالْآيَاتِ إِلَّا تَخْوِيفًا)
After Punctuation Remove	#هزه_ارضيه_المدينه_المنوره وَمَا نُرْسِلُ بِالْآيَاتِ إِلَّا تَخْوِيفًا
After diacritics remove	#هزه_ارضيه_المدينه_المنوره وما نرسل بالآيات إلا تخويفا
After special characters remove	هزه ارضيه المدينه المنوره وما نرسل بالآيات إلا تخويفا
Data Sample 2	RT Saleh88112 حزب الله يستهدف البحرين الداخلية البحرينية إيران تأوي عناصر إرهابية مسقطة جنسياتهم https://t.co/NkVkE0jV6z
After URL Remove	RT Saleh88112حزب الله يستهدف البحرين الداخلية البحرينية إيران تأوي عناصر إرهابية مسقطة جنسياتهم
After Latin Letters remove	حزب الله يستهدف البحرين الداخلية البحرينية إيران تأوي عناصر إرهابية مسقطة جنسياتهم
Normalize Alef and Teh	حزب الله يستهدف البحرين الداخليه البحرينيه ايران تاوي عناصر ارهابيه مسقطه جنسياتهم
Data Sample 3	عاااجل انباء عن الافراج عن مدين وصلاح نجلي الشهيد الزعيم علي عبدالله صالح
After Stop Words Remove	عاااجل انباء عن الافراج مدين وصلاح نجلي الشهيد الزعيم علي عبدالله صالح
After Repeated Letters remove	عاجل انباء عن الافراج مدين وصلاح نجلي الشهيد الزعيم علي عبدالله صالح

Table 2 Normalization Procedure Steps Examples

B. In the **tokenization** procedure the text is divided into basic units where the units could be paragraphs, sentences, words, or numbers [37]. Since the posts in twitter are short and contain at most 280 characters, posts are split into words directly. Each word of each post then is converted to integer value representing its index in a dictionary. This dictionary is a matrix built from the top 20000 word features in the dataset. This step is essential for the representation techniques that are based on word embedding approach, since the proposed models are implemented using Keras deep learning library; the embedding layer of Keras

accepts positive integer values only as input when using GloVe as feature representation technique in the Feature Representation phase.

2.2 Variable Length Post-series Construction

The main objective of this study is to detect rumors in Twitter's Arabic at the event level; as mentioned earlier, when an event or piece of news gains wide attention on Twitter, it is discussed through a series of tweets. Thus, the event or news will be dealt with as a whole during the classification process rather than independently classifying each tweet about a specific news or event as a rumor.

Each event in the dataset contains hundreds to thousands of tweets, and each post has a mutable number of words as well, which makes the events variable in length. The posts of each event are separated from each other but labeled by their event id and class id, which means that the dataset has to be prepared and the shape of the data instances has to be adjusted to feed them probably to the deep learning model as Events.

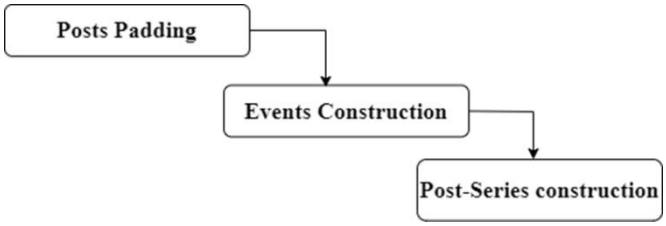


Figure 2 Variable Length Post-Series Construction

As shown in **Figure 2**, the following three steps would be achieved to reshape the dataset and construct the event samples:

- **Post padding:** In this step, each post in the dataset would be padded with 0s, so all the posts would be equal in length. The maximum length of a post is 70 words since it is the maximum number of words that can be used to compose a tweet containing 280 characters. This step is necessary for feature representation techniques that utilize an embedding layer while feeding the deep learning model.

- **Event Construction:** for each event, the related posts would be grouped in a matrix where each matrix represents a sample to be fed to the model; each matrix in the events matrices is linked to one item in the label's matrix representing the event class to be either (1for rumors) or (0 for non-rumor).

- **Post-Series Construction:** since each event matrix may contain hundreds to thousands of posts, each matrix would be divided into small chunks (series) to facilitate the model's feeding process. The maximum length would be set to N posts for each chunk. Each event should contain a minimum number of chunks (series) Min; if an event containing less N× Min would be padded with Zeros.

2.3 Feature Representation

As mentioned earlier, this study relies only on the tweet's latent textual as a linguistic feature to be extracted from tweet text directly and fed to the deep learning model. Latent textual features refer to text embedding into a representation that deep learning models can understand (numeric form). Textual features may be extracted at the word, sentence, and document levels. In this case, an event may be encoded in latent vectors, which are able to serve as the input for classifiers (such as SVMs) immediately or later incorporated into neural network architectures[11]. In this paper two different approaches have been used for feature representation: Word Embedding and TF-IDF sparse vectors.

A. Word Embedding

In deep learning, word embedding is the process of encoding textual items to real-valued vectors. It captures the relation between the different words of a large text corpus; words with similar semantic or syntactic meaning would have closer vector values and be clustered within the same space. In word embedding, those real-valued vectors are called dense vectors and represent the projection of the word into a continuous vector space. As mentioned earlier the models were implemented using Keras Tensorflow Library and this library offers an embedding layer that can be used to learn the word embeddings of the dataset. Word embedding was performed in two flavors. **First,** using the embedding layer alone, where the embedding is learned along with the model itself. **Secondly,** using the embedding layer with pre-trained embeddings where the layer used once to load pre-trained GolVe model and once Word2Vec model.

Glove Embeddings refers to Global Vectors, a global and multilingual word embedding algorithm based on unsupervised learning and developed by Stanford University. This algorithm along with other word embedding algorithms is used to train fixed-length continuous valued vectors utilizing large textual dataset. It is count-based algorithm that depends on the mathematical statics that measures word co-occurrences or how frequent two certain words would appear together. The GloVe miasmatical model attempts to investigate the word vector in a way that determines whether the dot product of those words equals the probability logarithmic value of these words appearing together or the probability of their co-occurring. It can be represented by the following equations:

$$w_i^T + \vec{w}_k + b_i + \vec{b}_k = \log(X_{i_k}) \quad (1)$$

$$f(X_{i_k}) = \begin{cases} \left(\frac{X_{i_k}}{xmax}\right)^{\alpha} ; if X_{i_k} < xmax \\ 1 ; lainnya \end{cases} \quad (2)$$

$$J = \sum_{i,k=1}^{v} f(X_{i_k})(w_i^T + \vec{w}_k + b_i + \vec{b}_k - \log(X_{i_k})) \quad (3)$$

The w represents the word vector while $\vec{w}$ stands for the context word vector, b_i and $\vec{b}_k$ would be the scalar biases for i-word and the k-word context respectively. Each X_{i_k} member in the word co-occurrence matrix X reflects how many times the word i appears in the k -word context. A context word is a group of words that appear before and after the word "i" in the same proportion as the specified window size. After then, a weighting will be assigned to each word using way of 1 /distance. The distance here refers to the distance that exists between the word's position and its context. Equation (2) calculates the function of the weight, Then by incorporating equations (1) and (2) into a cost function produce a model as seen in equation (3).

Word2Vec Embeddings Google released Word2vec, a natural language processing (NLP) technology, in 2013. It is a collection of related models used to generate word embeddings. These are shallow, two-layer neural networks that have been taught to recreate word linguistic contexts. Word2vec takes a huge corpus of text as input and outputs a vector space with several hundred dimensions, with each unique word in the corpus allocated a matching vector in the space. In this study, a pretrained word2vec model on Arabic Twitter corpus have been utilized to produces word embeddings; this model called AraVec. AraVec is an open source pre-trained distributed word representation project that seeks to provide free and strong word embedding models to the Arabic NLP research community. AraVec's first version includes six distinct word embedding models developed on top of three major Arabic content domains: Tweets, Wikipedia, and others. In this paper, the AraVec Skip-Gram Twitter version has been utilized.

Each word in the skip-gram model contains two-dimensional vector representations that could be utilized for computing conditional probabilities. As an illustration, assuming a word has the index i in the dictionary, then its two vectors being employed as a center word and a context word, respectively, are denoted by $v_i \in \mathbb{R}^d$and $u_i \in \mathbb{R}^d$. The conditional probability of generating any context word w_o (with index o in the dictionary) given the center word w_c (with index c in the dictionary) can be modeled by a softmax operation on vector dot products as in equation 5.where the vocabulary index set $V = \{0,1, \ldots\ldots, |V| - 1\}$ Given a text sequence of length T, where the word at time step t is denoted as $w^{(t)}$
Assume that context words are independently generated given any center word. For context window size m, the likelihood function of the skip-gram model is the probability of generating all context words given any center word as shown in equation 6.

$$P(w_o|w_c) = \frac{exp(u_o^T v_c)}{\sum_{i \in v} exp(u_i^T v_c)} \quad (5)$$

$$\prod_{t=1}^{T} \prod_{-m \le j \le m, j \ne 0} P\left(w^{(t+j)} \middle| w^{(t)}\right) \quad (6)$$

B. **TF-IDF sparse vectors** which stand for Term Frequency Inverse Document Frequency is a popular algorithm used widely to produce numerical representation for textual documents or datasets. It relies on statistically measured weights that reflect how important a word to a document within a collection of documents. TFIDF of a word is resulted from calculating the TF and IDF for the word, then multiplying those values together. TF represent how frequent a

target terms appears in a document, the ratio of target term number to the all terms number within this document. IDF in the other side is obtained by calculating the ratio of all documents number to the documents number in which the target term appears. Calculating the TF-IDF vector is represented mathematically as follow:

$$TF_{ij} = \frac{f_{ij}}{n_{ij}} \quad (4)$$

$$IDF_i = 1 + \log\left(\frac{N}{c_i}\right) \quad (5)$$

$$w_{ij} = TF_{ij} \times IDF_i \quad (6)$$

Where f_{ij} is the occurrences number of term i in tweet j while n_i represents total number of words the tweet j contain. In equation, N represents total number of tweets in the dataset while c_i stands for the number if tweets that contains the term$i$. Finally, equation (6) calculates the Tf-IDf score of the term i in tweet$j$.

2.4 Classification Models

In this paper, two classification Recurrent Neural Network(RNN)models have been employed for the Arabic event rumor detection task namely: Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (Bi-LSTM) in which each model have been fed with the previous phase output. In this section the both techniques would be described.

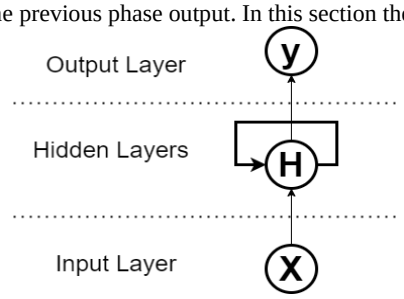


Figure 3 RNN basic structure

The basic structure of RNNs as seen in figure , consists of three main components (layers): First, an Input Layer which consists of artificial neurons that only accept data sequences and passes them to the hidden layers; Second, Hidden Layer that consists of one or multiple layers of artificial neurons designed to perform the necessary mathematical functions and computations for the learning process; and finally, an Output layer where the necessary computations take place to produce the classification result (prediction). In RNNs, the data sequences are fed recursively to the hidden layers, where the neurons contain a loop and internal memory; consequently, what has been learned from the previous input (hidden layer output) would be considered when processing the next input (next data sequence).

A. **Long Short-Term Memory (LSTM)**

LSTM was presented for the first time by Hochreiter and Schmidhuber [40] in 1997. LSTM is an extended Recurrent Neural Network (RNN) and has been discovered to overcome the vanishing gradient problem in traditional RNNs, which makes RNNs difficult to train [41]. Recurrent neural networks (RNNs) are highly capable of modelling sequential data[11].

The internal structure of the LSTM hidden layer consists of a cell state and three main gates, namely forget, input, and output gates. The input gate determines which information from the inputted data is to be stored in the cell state. In contrast, the forget gate determines which information is to be forgotten from the previous data. The output gate determines the output of the current cell state. LSTM networks are the optimal learning model to deal with long-distance input sequences, especially when the data sequences' patterns are changing over time. Consequently, it would be employed in detecting rumours at the event-level since every single event contains hundreds and thousands of tweets spreading and discussing a specific event rumour.

The architecture of a vanilla LSTM hidden layer is represented by Figure 4 and the following equations:

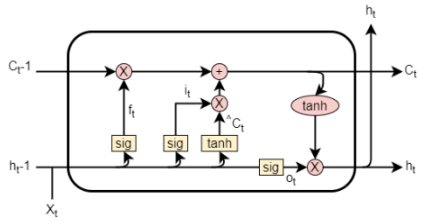


Figure 4 LSTM Architecture

$$i_t = \sigma(W_i.[x_t, h_{t-1}] + b_i) \quad (7)$$
$$f_t = \sigma(W_f.[x_t, h_{t-1}] + b_f) \quad (8)$$
$$\acute{C} = tanh(W_C.[x_t, h_{t-1}] + b_c) \quad (9)$$
$$C_t = f_t \odot c_{t-1} + i_t \odot \acute{C} \quad (10)$$
$$O_t = \sigma(W_o.[x_t, h_{t-1}] + b_o) \quad (11)$$
$$h_t = o_t \odot \tanh(c_t) \quad (12)$$

Figure depicts the LSTM's inputs and outputs for a single timestep, where i_t, f_t, O_t and C_t represent the input gate, forget gate, output gate and the cell state respectively. The x_t represents the current input sequence while h_{t-1} is the hidden state output of the previous timestep and c_{t-1} is the cell state of the previous timestep as well .Each gate uses point-wise multiplication with the sigmoid(σ) activation function operations to regulate the state of the memory cells while the weight and bias of any gate unit are indicated by W and b, respectively. The input gate is formulated by equation; the sigmoid activation function receives two inputs: the current input x_t and the previously hidden stateh_{t-1}. This gate determines which information is updated by converting the values from zero to one. The value one indicates significance, whereas zero indicates unimportance. Equation represents the forget gate; The sigmoid function transitions information from the now input x_t and the previously hidden state h_{t-1}. Consequently, the output value of the forget gate ranges from zero to one. The information will be eliminated if the value is very near zero. In contrast, the information will be retained if the value is closer to one. The new value is updated to the cell state once the cell state $\acute{C}_t$ has been determined by equation where the both the current input x_t and hidden state h_{t-1} are fed to the tanh function. The tanh is the hyperbolic tangent activation function, $\odot$ is the dot product operation between two vectors while C_t is the new memory cell and formulated by equation. As a final step, the output gate would determine the next hidden state as shown in equations and. The new hidden state h_t along with the new C_t would be passed to the next time step.

As depicted in Figure 5, the LSTM model developed in this paper is composed of the following layers: an input layer, embedding layer, LSTM layer, attention layer, and dense layer. The output of the previous phase is fed into the model as input through the Input Layer; in this layer, the input shape is initialized to be (K × N); N represents the number of posts in one data sample, where K represents the length of each post. This layer is connected to the embedding layer, which is responsible for converting the textual data tokens to a numeric vector to be fed probably to the LSTM layer; each post in each sample is converted to a numeric vector with the size E. Following the Embedding layer are two LSTM layers with 128 neurons for each layer to prevent model overfitting; dropout and recurrent dropout have been utilized. The next layer is the Attention Layer, where a Softmax attention mechanism has been employed to discriminate the most important features from the high duplication in the dataset. The last layer is the dense layer, where the output of the last layer neurons is fed to; this layer represents the output layer of the proposed model containing only one neuron and utilizes 'sigmoid' activation.

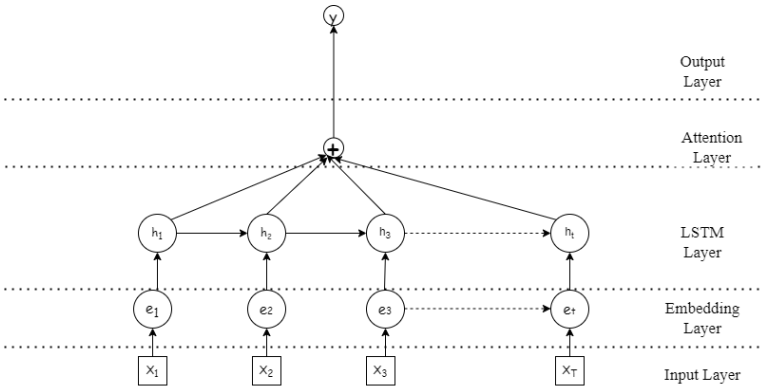


Figure 5 Long Short-Term Memory Model

B. **Bi-directional LSTM model**

A special version of the LSTM neural network in which the model learns in two directions: forward and backward. In traditional LSTM, input sequences flow in only one direction, either backward or forward. In contrast, according to the Bi-directional LSTM main architecture, the single Bi-LSTM layer consists of two separate LSTM layers, as shown in Figure 6; the first layer learns and examine the input sequence from the past to the future (forward), while the other layer learns from the future to the past (backward). The resulting outputs from both directions (layers) are stacked together, and the final hidden state results from concatenating the two hidden states of the two LSTM layers (forward and backward). BiLSTM architecture can be demonstrated by the equations where three sequences are computed: a forward hidden state $\vec{h}_t$, a backward hidden state $\overleftarrow{h}_t$, and the output hidden state h_t. The input sequence's past information would be obtained by the forward LSTM, whereas the backward LSTM may acquire the input sequence's future information. The output from both hidden states is then merged into one hidden stateh_t.

$$)\vec{h}_t = \mathcal{H}(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t+1} + b_{\vec{h}} \quad (13)$$

$$)\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}} \quad (14)$$

$$h_t = \vec{h}_t \oplus \overleftarrow{h}_t \quad (15)$$

While $\mathcal{H}$ can be any activation function, the symbol $\oplus$ stands for summation by component that utilized to combine the forward and backward output components. The proposed Bi-LSTM was constructed following the same layers' flow as the LSTM model, except that the two LSTM layers were replaced with two Bi-LSTM layers containing 128 neurons each.

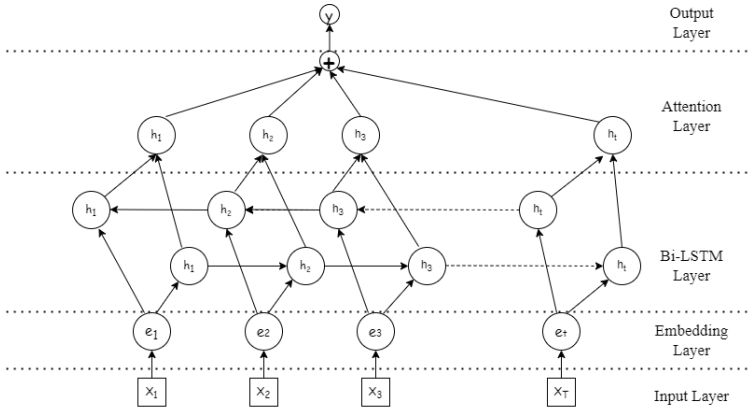


Figure 6 Bi-directional Long Short-Term Memory Model

C. The Attention Mechanism

Attention is a complex cognitive skill that humans require. One key aspect of perception is that people do not process complete information at once. Instead, humans selectively focus on a subset of information when and where it is required while ignoring other perceptible information. For example, while visually observing objects, individuals often do not see all of the scenery from beginning to end but instead, notice and pay attention to particular areas as needed [42]. When individuals discover that a scene frequently has something they want to observe in a specific part, they will learn to focus on that part when comparable scenarios arise again. They will pay more attention to the valuable part. This allows humans to swiftly select high-value information from huge amounts of data while using limited processing resources. The attention mechanism significantly improves the efficiency and accuracy of perceptual information processing [42]. The attention mechanism has become an increasingly prominent component of neural architectures and has been applied to various tasks, including image captioning, generation, and text classification. The utilized attention mechanism in this research was introduced by Zhou, et al. [43], which is a soft attention layer that is included between the LSTM (or Bi-LSTM) layer and the output(dense) layer and is expressed by the following equations:

$$\mathrm{M} = softmax(\mathrm{H}) \quad (16)$$

$$\propto = softmax(w^T M) \quad (17)$$

$$r = H \ \propto^T \quad (18)$$

Let H be a matrix consisting of output vectors $[h_1, h_2, \ldots, h_T]$ that the LSTM layer produced where T is the sentence length. The representation r of the sentence is formed by a weighted sum of these output vectors. where $H \in \mathbb{R}^{d^w \times T}$, d^w is the dimension of the word vectors, ω is a trained parameter vector and w^T is a transpose. The dimension of $w, \propto, r$ is d^w, T, d^w separately. The final sentence-pair representation used for classification is obtained from:

$$h^* = \tanh(r) \quad (19)$$

2.5 The Dataset

In this paper, a public dataset would be used for both the baseline model and the proposed models. The dataset have been collected and published by [15] to be publicly available for future researches. A collection of Arabic rumor and non-rumor tweets have been collected using Twitter Search API. The collected tweets are related to specific rumor and non-rumor event; Rumor event was chosen from anti-rumors authority (http://www.norumors.net/) and Ar-Riyadh daily journal (http://www.alriyadh.com/). The authors collected 271,000 tweets that are divided to 89 events of rumor and 88 events of non-rumor. For each event, the original tweets have been collected and two types of features have been extracted namely Tweet-based features and Topic-based features. Two aspects are worth mentioning: firstly, in this study, only the tweet content was used as a feature since this study depends only in the linguistic feature that is extracted directly from the tweet text itself. Secondly, only 100,000 tweets that represent almost 70 events divided between rumor and non-rumor events have been utilized depending in the available computation resources and memory to run the model.

2.6 Performance Analysis Metrics

The proposed model's effectiveness and performance is evaluated utilizing the four main performance metrics: The Accuracy, Precision, Recall and F1-Score. Those metrics are calculated using the confusion matrix as shown in Table 3.

		Actual	
		Positive	Negative
Predicted	Negative	True Positive (TN)	False Positive (FP)
	Positive	False Negative (FN)	True Negative (TN)

Table 3 the confusion matrix

Considering that "Positive" represents the class "Rumor" and "Negative" represents the class "non-Rumor", the confusion matrix's Table, as shown below, represents four combinations of the predicted and actual values of the test dataset instances:

True Positive: count of Rumours that were correctly predicted as Rumours.
False Positive: count of non-Rumours that were mistakenly predicted as Rumour.
True Negative: count of non-Rumours that were correctly predicted as non-Rumour.
False Negative: count of Rumours that were mistakenly predicted as non-Rumour.

It should be clarified that there is only one way to calculate the accuracy where the Precision along with the Recall and F1 can be calculated in two different methods: depending on distribution (balanced or imbalanced) and the importance of the classes:

A. **Weighted average** considers the weight of each class when calculating the metric average, considering the number of instances of each class in the dataset. It is usually used with imbalanced data to assign greater contributions to classes that have higher number of instances.
B. **Macro-average** calculates the metric individually for each class and then calculates the metric average, thus treating all classes equally; it is best to use balanced data.

The performance metrics is calculates as illustrated below:

Accuracy is the closeness of a measured value to a standard or known value. It represents the ratio of true positive and negative instances to the total data set instances. In other words, it represents the percentage of test data instances or events correctly classified.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (20)$$

Precision This metric assesses the accuracy of the classification model in categorizing a dataset instance as positive In other words; it is the quality of a positive prediction made by the model. The weighted average precision is calculated using the following equations:

$$precision_{positive} = \frac{TP}{TP+FP} \quad (21)$$

$$precision_{negative} = \frac{TN}{TN+FN} \quad (22)$$

$$macro - precision = \frac{precision_{positive}+precision_{negative}}{2} \quad (23)$$

$$weighted - precision = \frac{precision_{positive}*N_{positive} + precision_{negative}*N_{negative}}{N_{positive} + N_{negative}} \quad (24)$$

Recall is the rate of the true positives and called also the sensitivity, which is the true positives divided by the true positives plus the false negatives. In this research it represents ratio of the accurately predicted rumor tweets to the total number of actual rumor tweets. The recall is calculated using the following equations:

$$recall_{positive} = \frac{TP}{TP+FN} \quad (25)$$

$$recall_{negative} = \frac{TN}{TN+FP} \quad (26)$$

$$macro - recall = \frac{recall_{positive}+recall_{negative}}{2} \quad (27)$$

$$weighted - recall = \frac{recall_{positive}*N_{positive} + recall_{negative}*N_{negative}}{N_{positive} + N_{negative}} \quad (28)$$

F-Score is the combination of Precision and recall, giving the balanced evaluation between both Precision and Recall. F1-score would be calculated with the following equation for both macro and weighted average:

$$F1 - score = 2 * \frac{precision*recall}{precision+recall} \quad (29)$$

3. RESULTS AND DISCUSSION

In this paper, a new framework was built for Arabic rumor detection on Twitter with the ability to detect rumor in the event level. Two deep learning techniques have been employed in this paper namely: LSTM and BiLSTM as introduced earlier. Google Colab notebooks were used to implement the model utilizing Tensorflow Keras library. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. In this section, the performed experiments on the introduced models would be clarified then the results of each experiment would be discussed. The experiments have achieved in this research in two phases as described and discussed below.

A. Experiments Phase 1

The aim behind the first phase of experiments was to test and examine the developed models and the proposed framework before applying more experiments and enhancements. In this phase the training and tuning process of those models was controlled by the following hyper parameters:

Activation	Sigmoid	**Epochs**	10
Loss Function	binary_crossentropy	**Batch Size**	128
Optimizer	adam	**Random State**	42
Dropout	0.5	**Recurrent Dropout**	0.8

Table 4LSTM and Bi-LSTM Model's Hyperparameters

Two main experiments were performed on each one of deep learning models:

a. Attention mechanism where the models were built once with a softmax attention embedded in the model and once without the attention; this would measure the extent to which the attention affects the performance of the model.
b. Feature representation where the models have been experimented with three different techniques of feature representation:
   - TF-IDF where the TF-IDF vectorizer offered by the scikit-learn library has been utilized and the maximum feature value has been set to 500.
   - Keras Embedding layer with pre-trained GloVe embedding where the maximum feature value was set to 20000 and the vector size to 100.
   - Keras Embedding layer where the embedding is learned along with the model itself and the vector size =100.

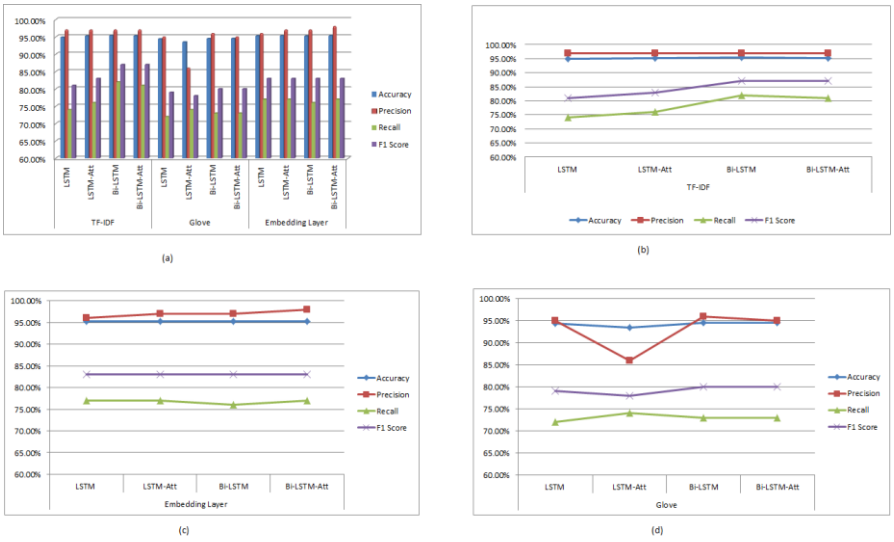


Figure 7 Phase 1 Experiments results

Figure 7 summarizes the performance results of the two models. Analyzing the results, it can be noted that:

- In terms of Accuracy and precision, Bi-LSTM-TFIDF and Bi-LSTM-Attention-Embedding-layer models achieved the best results, outperforming the remaining.

- Employing the embedding layer alone outperforms the GloVe embedding results for both models; the reason behind this that the embedding layer learns the embedding from the training dataset where its context will be more closer to the test set than the pre-trained embedding.

- The attention mechanism enhanced the performance of the models in general but fails when experimenting with the Glove embedding for the LSTM model.

- TF-IDF as well outperforms the performance of the other representations, In fact, TF-IDF creates vectors based on the importance of the word to the corpus while GloVe creates vectors based on the occurrences of the word in a document; this may demonstrate the reason for the higher accuracy resulted from the TF-IDF representation, but the comparison between TF-IDF and Glove may not be fair since the values of the maximum features are not equal. The process of producing the TF-IDF vectors consumes a huge amount of memory, so the number of the features had to be minimized.

- What has been observed during the time the models were running was that the LSTM model were faster and consumed fewer computation resources (RAM and CPU) than the Bi-LSTM model.

- Regarding the consumed resources as well, it has been noticed that TF-IDF requires more computation resources as well compared to the other feature representations techniques.

B. Experiments Phase 2

In this phase more experiments were conducted and the hyperparameters of the developed models were optimized; the training and tuning process of those models was controlled by the following hyper parameters:

Activation	Sigmoid	**Epochs**	20
Loss Function	binary_crossentropy	**Batch Size**	64
Optimizer	adam	**Random State**	42
Dropout	0.5	**Recurrent Dropout**	0.8

Table 5 LSTM and Bi-LSTM Model's Hyperparameters

Three main experiments were performed on each one of deep learning models:

A. Attention mechanism where each model was built once with a softmax attention embedded in the model and once without the attention.
B. The Training size was set to 50%, then 60%, then at last to 70%.

C. Feature representation where four different techniques for feature representation were used:
   - TF-IDF where the TF-IDF vectorizer offered by the scikit-learn library has been utilized and the maximum feature value has been set to 700.
   - Keras Embedding layer with pre-trained GloVe embedding where the maximum feature value was set to 20000 and the vector size to K=100.
   - Keras Embedding layer with pre-trained Word2Vec embedding where the maximum feature value was set to 20000 and the vector size to K= 100.
   - Keras Embedding layer where the embedding is learned along with the model itself and the vector size K =100.

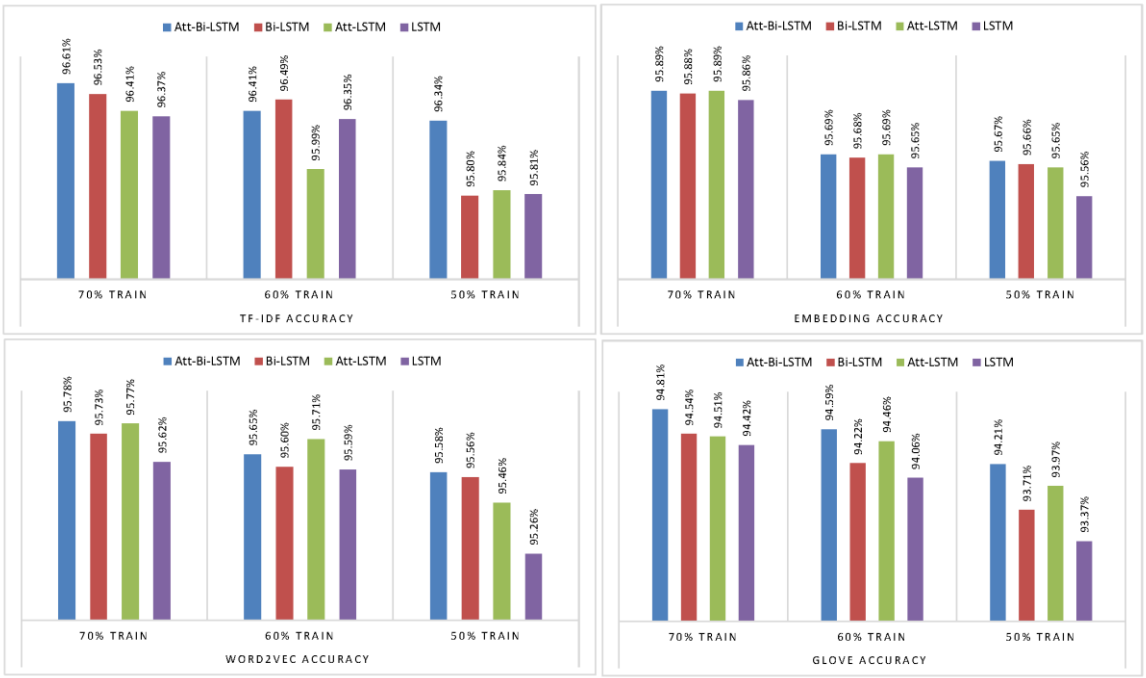


Figure 8Accuracy Results

Figure 8 shows the accuracy results of all the archived experiments on the two models; LSTM and Bi-LSTM. Regarding the addition of the attention mechanism, the results emphasize that the soft attention had a positive effect on both models in almost all of their versions and based on the overall performance, as there was a noticeable increase in the accuracy. The training set's size as shown, affected the models' performance, where the accuracy decreased each time the training size was decreased. Gholamy, et al. [44] state that when there is a model for a physical phenomenon with multiple unknown parameters. These parameters must be derived from known observations; this process is known as training the model. In statistics, the more data points to be utilized, the more accurate the final estimations.[44]. They also emphasized that the optimal amount of data provided for training is between 80% and 70% of the overall data volume, compared with 20% to 30% for testing. Any other ratio will have a detrimental impact on the model's performance. Regarding the feature representation techniques, TF-IDF outperforms the other representation techniques, achieving accuracy above 96% among all experiments of both models, while Keras Embedding layer, word2vec and Glove achieved 95.89%, 95.78% and 94.81% respectively at their best. TF-IDF creates vectors based on the importance of the word to the corpus; on the other hand, Word2vec generates vectors that reflect the word context, and GloVe generates vectors that represent word co-occurrence in the document. Those facts demonstrate the reason for the higher accuracy resulting from the TF-IDF and Word2Vec representations. Utilizing the Keras embedding layer by itself to generate the text features achieves better results than Word2Vec and GloVe. This is not surprising since this layer trained to generate vectors directly from the dataset, which results in capturing more semantics and generating vectors that are more relevant to the dataset context. This encourages the training of additional embedding models on datasets derived from Arabic rumor content on Twitter in order to obtain better results that may outperform the efficiency of the TF-IDF models. It has been observed that training the Bi-LSTM model is slower than training the LSTM model since the Bi-LSTM model trains in two directions (forward and backwards) and requires fetching extra batches of data to achieve the equilibrium, as reported by [45]. It has also been noted that the Bi-LSTM gives an impact and accuracy level close to the accuracy level of adding an attention layer to the LSTM model. There are certain extra data features that BiLSTM may capture that traditional LSTM model cannot expose since their training is limited to one direction.

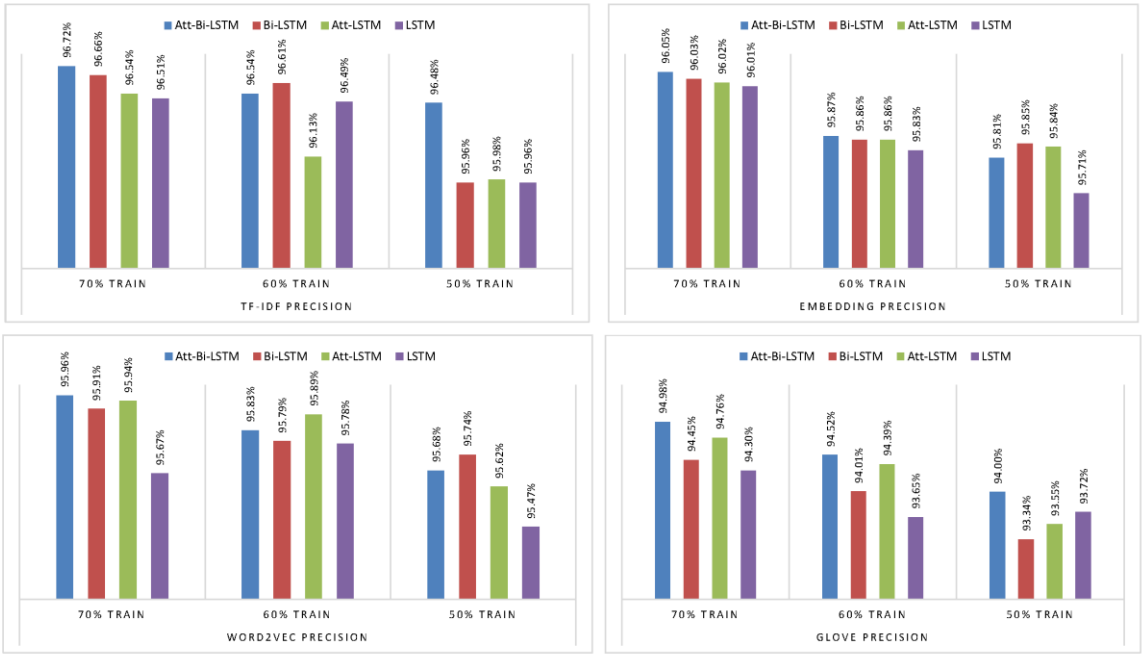


Figure 9 Precision Results

The above figure shows the weighted-average precision results. This metric is supposed to ensure that what has been classified as rumor or non-rumor is indeed rumor or non-rumor; it measures the ratio of rumors or non-rumor that have been correctly classified. It can be observed that in most of the situations, except for GloVe, this metric is higher than the accuracy; this implies that the proposed models will often be precise in predicting both classes (rumor and non-rumor), even if the accuracy value is far from the standard value.

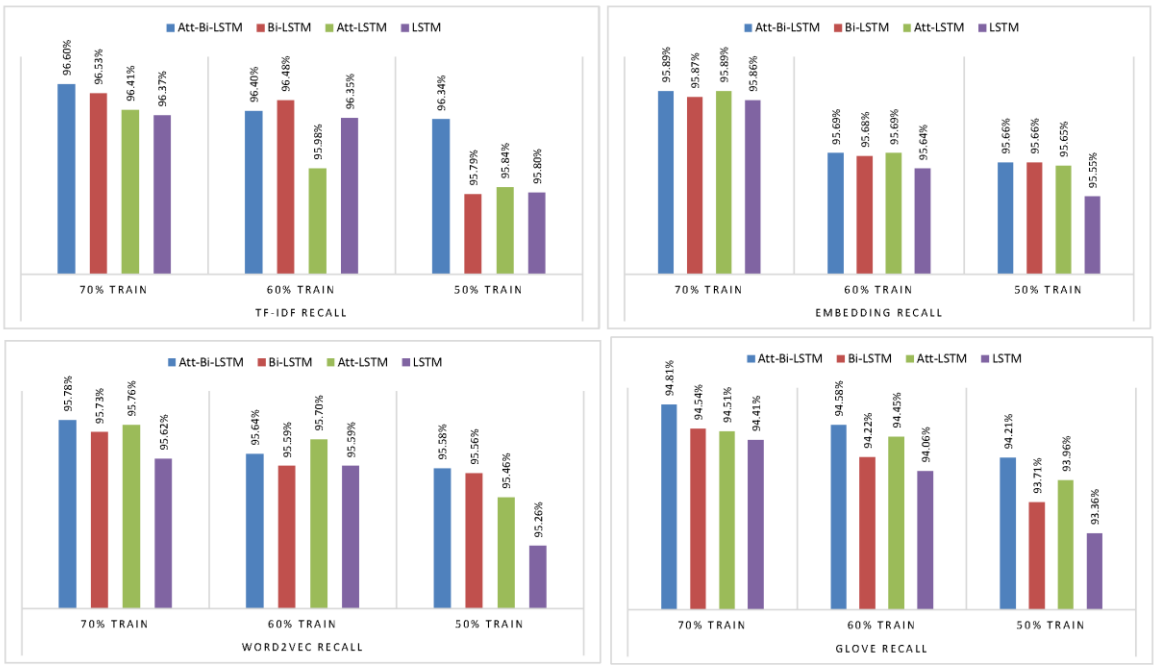
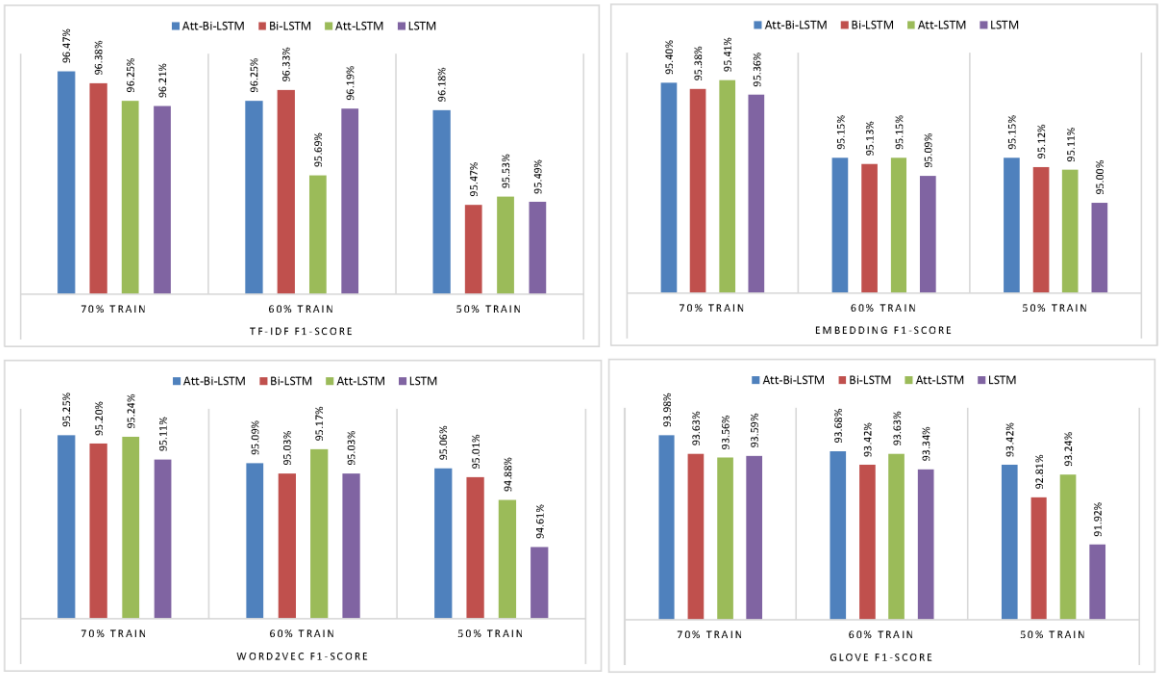


Figure 10 Recall Results

The average of the Recall is designed to evaluate each model's sensitivity to both classes (rumor and non-rumor). It can be noted from the above figure that the recall values are very close or equal to the related accuracy values. This implies the models are somehow "balanced", that is, their ability to correctly classify Rumors events is same as their ability to correctly classify non-rumor events. In other word they have the same high level of senstivity towards the both classes.



After calculating the recall and precision average, the F1 score combines two competing metrics scores of each model. Note that maximizing the F1 score implies simultaneously maximizing both precision and recall, which is a balance between them. It can be concluded that Bi-LSTM combined with the TFIDF, in terms of accuracy and recall; is the best choice for identifying Arabic event rumors on Twitter.

4. CONCLUSION

In this paper, two deep learning models have been employed to detect Arabic rumours on Twitter content: LSTM and Bi-LSTM. The main objectives behind this study were to construct deep learning detection models capable to detect rumours based on event-level relaying on latent textual features only (linguistic features), then to enhance the performance of the models by embedding an attention mechanism. The two models have been experimented with four feature representation techniques namely: TF_IDF, Keras embedding layer, word2vec, and GloVe. The employed attention mechanism noticeably improved the resulting performance from all the constructed versions of the models, the TF-IDF feature representation outperform the other techniques. The Bi-LSTM model versions achieves better results than LSTM versions, due to its ability to learn in both directions. In future work, other features besides the Textual features may be examined and a large amount of data will be collected from Twitter to build a more accurate embedding models dedicated to the deep learning techniques for detecting rumours in Arabic content.

REFERENCES

[1] M. Hussein, A. Abu Issa, M. Washha, I. Elayyan, W. Makho , and I. Shneneh, "Arabic Rumor Detection using Collaborative Framework over Social Networks and Web," *Journal of Computer Science*, 2018.

[2] T. Chen, X. Li, H. Yin, and J. Zhang, "Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2018, pp. 40-52.

[3] N. DiFonzo and P. Bordia, "Rumor, gossip and urban legends," *Diogenes*, vol. 54, pp. 19-35, 2007.

[4] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," 2018.

[5] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences,* vol. 497, pp. 38-55, 2019.
[6] S. Vosoughi, "Automatic detection and verification of rumors on Twitter," Massachusetts Institute of Technology, 2015.
[7] K. M. d. I. Treen, H. T. Williams, and S. J. O'Neill, "Online misinformation about climate change," *Wiley Interdisciplinary Reviews: Climate Change,* vol. 11, p. e665, 2020.
[8] L. Wu, F. Morstatter, K. M. Carley, and H. Liu, "Misinformation in social media: definition, manipulation, and detection," *ACM SIGKDD explorations newsletter,* vol. 21, pp. 80-90, 2019.
[9] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, and D. Rothschild, "The science of fake news," *Science,* vol. 359, pp. 1094-1096, 2018.
[10] X. Zhou and R. Zafarani, "Fake news: A survey of research, detection methods, and opportunities," *arXiv preprint arXiv:1812.00315,* 2018.
[11] L. Hu, S. Wei, Z. Zhao, and B. Wu, "Deep learning for fake news detection: A comprehensive survey," *AI Open,* vol. 3, pp. 133-155, 2022/01/01/ 2022.
[12] S. Han, "Context-aware message-level rumour detection with weak supervision," University of Sheffield, 2020.
[13] R. M. B. Al-Eidan, H. S. Al-Khalifa, and A. S. Al-Salman, "Measuring the credibility of Arabic text content in Twitter," in *2010 Fifth International Conference on Digital Information Management (ICDIM)*, 2010, pp. 285-291.
[14] S. F. SABBEH and S. Y. BAATWAH, "ARABIC NEWS CREDIBILITY ON TWITTER: AN ENHANCED MODEL USING HYBRID FEATURES," *Journal of Theoretical & Applied Information Technology,* vol. 96, 2018.
[15] S. M. Alzanin and A. M. Azmi, "Rumor detection in Arabic tweets using semi-supervised and unsupervised expectation–maximization," *Knowledge-Based Systems,* vol. 185, p. 104945, 2019.
[16] M. Alkhair, K. Meftouh, K. Smaïli, and N. Othman, "An Arabic Corpus of Fake News: Collection, Analysis and Classification," in *International Conference on Arabic Language Processing*, 2019, pp. 292-302.
[17] G. Jardaneh, H. Abdelhaq, M. Buzz, and D. Johnson, "Classifying Arabic tweets based on credibility using content and user features," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2019, pp. 596-601.
[18] R. El Ballouli, W. El-Hajj, A. Ghandour, S. Elbassuoni, H. Hajj, and K. Shaban, "CAT: Credibility analysis of Arabic content on Twitter," in *Proceedings of the Third Arabic Natural Language Processing Workshop*, 2017, pp. 62-71.
[19] N. Hassan, W. Gomaa, G. Khoriba, and M. Haggag, "Credibility detection in twitter using word n-gram analysis and supervised machine learning techniques," *International Journal of Intelligent Engineering and Systems,* vol. 13, pp. 291-300, 2020.
[20] H. Mubarak and S. Hassan, "Arcorona: Analyzing arabic tweets in the early days of coronavirus (covid-19) pandemic," *arXiv preprint arXiv:2012.01462,* 2020.
[21] S. Alqurashi, B. Hamoui, A. Alashaikh, A. Alhindi, and E. Alanazi, "Eating garlic prevents COVID-19 infection: Detecting misinformation on the Arabic content of Twitter," *arXiv preprint arXiv:2101.05626,* 2021.
[22] 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.
[23] M. Al-Yahya, H. Al-Khalifa, H. Al-Baity, D. AlSaeed, and A. Essam, "Arabic fake news detection: comparative study of neural networks and transformer-based approaches," *Complexity,* vol. 2021, 2021.
[24] S. Alharbi, K. Alyoubi, and F. Alotaibi, "Deep Learning Based Rumor Detection for Arabic Micro-Text," *International Journal of Computer Science & Network Security,* vol. 21, pp. 73-80, 2021.
[25] A. Gumaei, M. S. Al-Rakhami, M. M. Hassan, V. H. C. D. Albuquerque, and D. Camacho, "An Effective Approach for Rumor Detection of Arabic Tweets Using eXtreme Gradient Boosting Method," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.,* vol. 21, p. Article 7, 2022.
[26] L. Wu, J. Li, X. Hu, and H. Liu, "Gleaning wisdom from the past: Early detection of emerging rumors in social media," in *Proceedings of the 2017 SIAM international conference on data mining*, 2017, pp. 99-107.
[27] A. Zubiaga, M. Liakata, and R. Procter, "Learning reporting dynamics during breaking news for rumour detection in social media," *arXiv preprint arXiv:1610.07363,* 2016.

[28] G. Liang, W. He, C. Xu, L. Chen, and J. Zeng, "Rumor identification in microblogging systems based on users' behavior," *IEEE Transactions on Computational Social Systems*, vol. 2, pp. 99-108, 2015.
[29] 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, p. 102018, 2020.
[30] J. P. Singh, N. P. Rana, and Y. K. Dwivedi, "Rumour Veracity Estimation with Deep Learning for Twitter," in *International Working Conference on Transfer and Diffusion of IT*, 2019, pp. 351-363.
[31] A. Aker, A. Sliwa, F. Dalvi, and K. Bontcheva, "Rumour verification through recurring information and an inner-attention mechanism," *Online Social Networks and Media*, vol. 13, p. 100045, 2019.
[32] W. Chen, Y. Zhang, C. K. Yeo, C. T. Lau, and B. S. Lee, "Unsupervised rumor detection based on users' behaviors using neural networks," *Pattern Recognition Letters*, vol. 105, pp. 226-233, 2018.
[33] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," 2017.
[34] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," 2016.
[35] M. K. Jain, D. Gopalani, Y. K. Meena, and R. Kumar, "Machine Learning based Fake News Detection using linguistic features and word vector features," in *2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, 2020, pp. 1-6.
[36] A. AlAttas, H. A. Mogaibel, and M. S. BinWahlan, "Event-Based Rumor Detection using LSTM Models For Arabic Content on Twitter," in *2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, 2022, pp. 1-7.
[37] R. Elbarougy, G. Behery, and A. El Khatib, "A Proposed Natural Language Processing Preprocessing Procedures for Enhancing Arabic Text Summarization," in *Recent Advances in NLP: The Case of Arabic Language*, ed: Springer, 2020, pp. 39-57.
[38] R. A. Salama, A. Youssef, and A. Fahmy, "Morphological word embedding for Arabic," *Procedia Computer Science*, vol. 142, pp. 83-93, 2018.
[39] O. S. Bahakam, M. S. F. Binwahlan, and H. A. Mogaibel, "Statistical Features and PageRank Scoring Fusion for Arabic Text Summarization," in *2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, 2022, pp. 1-8.
[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735-1780, 1997.
[41] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, pp. 5929-5955, 2020.
[42] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48-62, 2021.
[43] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, *Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification*, 2016.
[44] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation," 2018.
[45] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *2019 IEEE International conference on big data (Big Data)*, 2019, pp. 3285-3292.