Computers, Materials & Continua
DOI:10.32604/cmc.2022.021449
Article

Tech Science Press

Arabic Fake News Detection Using Deep Learning

Khaled M. Fouad[1,3], **Sahar F. Sabbeh**[1,2,*] **and Walaa Medhat**[1,3]

[1]Faculty of Computers & Artificial Intelligence, Benha University, Egypt
[2]University of Jeddah, College of Computer Science and Engineering, Jeddah, 21493, Saudi Arabia
[3]Information Technology and Computer Science, Nile University, Egypt
*Corresponding Author: Sahar F. Sabbeh. Email: sfsabbeh@uj.edu.sa

Received: 03 July 2021; Accepted: 14 October 2021

Abstract: Nowadays, an unprecedented number of users interact through social media platforms and generate a massive amount of content due to the explosion of online communication. However, because user-generated content is unregulated, it may contain offensive content such as fake news, insults, and harassment phrases. The identification of fake news and rumors and their dissemination on social media has become a critical requirement. They have adverse effects on users, businesses, enterprises, and even political regimes and governments. State of the art has tackled the English language for news and used feature-based algorithms. This paper proposes a model architecture to detect fake news in the Arabic language by using only textual features. Machine learning and deep learning algorithms were used. The deep learning models are used depending on conventional neural nets (CNN), long short-term memory (LSTM), bidirectional LSTM (BiLSTM), CNN+LSTM, and CNN + BiLSTM. Three datasets were used in the experiments, each containing the textual content of Arabic news articles; one of them is real-life data. The results indicate that the BiLSTM model outperforms the other models regarding accuracy rate when both simple data split and recursive training modes are used in the training process.

Keywords: Fake news detection; deep learning; machine learning; natural language processing

1 Introduction

The rise of social networks has considerably changed the way users around the world communicate. Social networks and user-generated content (UGC) are examples of platforms that allow users to generate, share, and exchange their thoughts and opinions via posts, tweets, and comments. Thus, social media platforms (i.e., Twitter, Facebook, etc.) are considered powerful tools through which news and information can be rapidly transmitted and propagated. These platforms empowered their significance to be the essence of information and news source for individuals through the WWW [1]. However, social media and UGC platforms are a double-edged sword. On the one hand, they allow users to share their experiences which enriches the web content. On the other hand, the absence of content supervision may lead to the spread of false information intentionally or unintentionally [2], threatening the reliability of information and news on such platforms.

False information can be classified as intention-based or knowledge-based [3]. Intention-based can be further classified into misinformation and disinformation. Misinformation is an unintentional share of false information based on the user's beliefs, thoughts, and point of view. Whereas the intentional spread of false information to deceive, mislead, and harm users are called disinformation. Fake news is considered disinformation as they include news articles that are confirmed to be false/deceptive published intentionally to mislead people. Another categorization of false information [4] was based on the severity of its impact on users. Based on this study, false information is classified as a) fake news, b) biased/inaccurate news, and c) misleading/ambiguous news. Fake news has the highest impact and uses tools such as content fabrication, propaganda, and conspiracy theories [5]. Finally, biased content is considered to be less dangerous and mainly uses hoaxes and fallacies. The last group is misleading news, which has a minor impact on users. Misleading content usually comes in the forms of rumors, clickbait, and satire/sarcasm news. Disinformation results in biased, deceptive, and decontextualized information based upon which emotional decisions are made, impulsive reactions, or stopping actions in progress. Disinformation results negatively impact users' experience and decisions, such as online shopping and stock markets [6].

The bulk of researches for fake news detection are based on machine learning techniques [7]. Those techniques are feature-based, as they require identifying and selecting features that can help identify any piece of information/text's fakeness. Those features are then fed into the chosen machine learning model for classification. In various languages, deep learning models [8] have recently proven efficiency in text classification tasks and fake news detection [9]. They have the advantage that they can automatically adjust their internal parameters until they identify the best features to differentiate between different labels on their own. However, no researches use deep learning models [10] for fake news detection for the Arabic language, as far as we know from the literature.

The problem of Fake news detection can have harmful consequences on social and political life. Detecting fake news is very challenging, mainly when applied in different languages than English. The Arabic language is one of the most spoken languages over the globe. There are a lot of sources for news in Arabic, including official news websites. These sources are considered the primary source of Arabic datasets. Our goal is to detect rumors and measure the effect of fake news detection in the middle east region. We have evaluated many algorithms to achieve the best results.

The work's main objective is exploring and evaluating the performance of different deep learning models in improving fake news detection for the Arabic language. Additionally, compare deep learning performance with the traditional machine learning techniques. Eight machine learning algorithms with cross-fold validation are evaluated, including probabilistic and vector space algorithms. We have also tested five combinations of deep learning algorithms, including CNN and LSTM.

The paper is organized as follows. Section 2 tackles the literature review in some detail. The proposed model architecture is presented in Section 3. Section 4 presents the experiments and the results with discussion. The paper is concluded in Section 5.

2 Literature Review

There are many methods used for fake news detection and rumor detection. The methods include machine learning and deep learning algorithms, as illustrated in the following subsections.

2.1 Fake News Detection Methods

Fake news detection has been investigated from different perspectives; each utilized different features for information classification. These features included linguistic, visual, user, post, and network-based features [5,11]. The linguistic-based methods tried to find irregular styles within text based on a set of features such as the number of words, word length, multiple words frequencies; unique word count, psycho-linguistic features, syntactic features (i.e., TF-IDF, question marks, exclamation marks, hash-tags . . . etc.) to discriminate natural and fake news [11]. Visual-based systems attempted to identify and extract visual elements from fictitious photos and movies [12] by using deep learning approaches. The user-based methods analyzed user-level features to identify likely fake accounts. It is believed that fake news can probably be created and shared by fake accounts or automatic pots created for this sake. User-based features were used to evaluate source/author credibility. Those features include, among others: the number of tweets, tweet repetition, number of followers, account age, account verifiability, user photo, demographics, user sentiment, topically relevancy, and physical proximity [13,14]. The post-based methods analyzed users' feedback, opinions, and reactions as indicators of fake news. These features included comments, opinions, sentiment, user rating, tagging, likes, and emotional reactions [14]. The network-based methods of social networks enabled the rapid spread of fake news. These methods tried to construct and analyze networks from different perspectives. Friendship networks, for instance, explored the user followers relationship. In comparison, stance networks represent post-to-post similarities. Another type is the co-occurrence networks, which evaluate user–topic relevancy [14].

Many methods considered rumors or fake news detection as a classification issue. These methods aim to associate attributes' values, like a rumor or non-rumor, true or false, or fake or genuine, with a specific piece of text. Researchers had utilized machine-learning methods, accomplishing optimistic results. Substitutionary researchers utilized other methods based on data mining techniques. They depended on extrinsic resources, like knowledge bases, to forecast either the included class of social media content or to examine their truthfulness. Many methods that detect rumors have concentrated on utilizing content features for classification. Few methods for rumor detection have depended on social context. Otherwise, rumor detection and verification methods predominantly utilized a combination of content and context features. Using this combination is since using the social context of rumors may significantly enhance detection performance [14]. Some different method categories of rumor detection that may be considered in the works of rumor detection analysis are shown in Tab. 1. These methods can be categorized into classification methods and other methods.

Table 1: Different attributes that participate in the start and growth of rumors on social media

Categories	Methods	Samples
User–related attributes	Machine learning	SVM, Random Forest, Decision Tree, Logistic Regression, Conditional Random Field (CRF), Hidden Markov Model (HMM).
Content–related attributes	Deep learning	Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN)
Other methods		Retweet Behavior, Diffusion Patterns, Anolamy Detection, Hawkes process, Crowdsourcing, Computational Fact-Checking

2.2 Machine Learning-Based Fake News Detection

Most of the fake news detection works to formulate the problem as a binary classification problem. The literature research may fall under the umbrella of three main classes [15]; feature-based machine learning approaches, networking approaches, and deep learning approaches. Feature-based machine learning approaches aim at learning features from data using feature-engineering techniques before classification. Textual, visual, and user features are extracted and fed into a classifier, then evaluated to identify the best performance given those sets of features. The most widely used supervised learning techniques include logistic regression (LR) [16], ensemble learning techniques such as random forest (RF) and adaptive boosting (Adaboost) [16–18], decision trees [18], artificial neural networks [18], support vector machines(SVM) [16,18], naïve Bayesian (NB) [16,18,19] and k-nearest neighbor (KNN) [16–18] and linear discriminant analysis (LDA) [16,20,21]. However, feature-based machine learning models suffer the issue of requiring feature engineering ahead of classification, cumbersome. Networking approaches evaluate user/author credibility by extracting features such as the number of followers, comments/replies content, timestamp, and using graph-based algorithms to analyze structural social networks [22,23].

2.3 Deep Learning for Fake News Detection

Deep learning (DL) [9,15] approaches use deep learning algorithms to learn features from data without feature engineering during training automatically. Deep learning models have proven a substantial performance improvement and eliminated the need for the feature extraction process. As stated earlier, deep learning models can automatically overcome the burden of feature engineering steps and automatically use training data to identify discriminatory features for fake news detection. Deep learning models showed remarkable performance in text classification general tasks [24–27] and have been widely used for fake news detection [28]. Due to their efficiency for text classification tasks, deep learning models have been applied for fake news detection from the NLP perspective using only text. For instance, in [29], deep neural networks were used to predict real/fake news using only the news's textual content. Different machines and deep learning algorithms were applied in this work, and their results showed the superiority of Gated Recurrent Units (GRUs) over other tested methods.

In [30], text content was preprocessed and input to recurrent neural networks (RNN), GRU, Vanilla, and Long Short Term Memories (LSTMs) for classifying fake news. Their results showed that LSTM follows GRU's best performance and finally comes vanilla. Text-based only classification in [31] using tanh-RNNs and LSTM with a single hidden layer, GRU with one hidden layer, and enhanced with an extra GRU hidden layer. In [32], bidirectional LSTMs together with Concurrent neural networks (CNN) were used for classification. The bi-directional LSTMs considered contextual information in both directions: forward and backward in the text. In [33], the feasibility of applying deep learning architecture of CNN with LSTM cells for text-only – based classification of RNN and LSTMs and GRUs were used in [34] for text-based classification. Deep CNN multiple hidden layers were used in [35].

Other attempts were made using text and one or more of other features in [36] Arjun et al. utilized ensemble-based CNN and (BiLSTM) for multi-label classification of fake news based on textual content and features related to the source's behavior speaker. In their model, fake news is assigned to one of six classes of fake news. (Pants-fire, False, Barely-true, Half-true, Mostly-true, and True). In [37], neural ensemble architecture (RNN, GRN, and CNN) and used content-based and author-based features to detect rumors on Twitter. Textual and propagation-based features were used in [38,39] for classification: the former used RNN, CNN, and recursive neural networks (RvNN). CNN and

RNN used only text and sentiment polarity of tweet's response, and the RvNN model was used on the text and the propagation. At the same time, the latter study constructed RvRNN based on top-down and bottom-up tree structures. These models were tested compared to the traditional state-of-the-art models such as decision tree (Tree-based Ranki DTR, Decision-Tree Classifier DTC), RFC, different variations of SVM, GRU-RNN. Their results showed that TD-RvRNN gave the best performance.

Post-based, together with user-based features, were used for fake news predictions [40]. They applied RNN, LSTMs, and GRUs and found out that LSTMs outperformed the other two models. In [41], text, user-based, content-based features, signal features were used for prediction tasks using a hierarchical recurrent convolutional neural network. Their experiments included (Tree-based Ranki DTR, Decision-Tree Classifier DTC), SVM, GRUs, BLSTMs. Tab. 2 summarizes the surveyed works in the literature.

Table 2: The summarization of the related works

Study	Used features	Experimented models	Dataset
Umer et al. [29]	Only text	Logistic regression Two-layer FNN GRUs Bidirectional RNN with LSTMs CNN with MaxPooling Attention-Augmented CNN	size: 63,000 articles.
Girgis et al. [30]	Only text	RNN (vanilla, GRU) LSTMs.	LIAR Dataset
Jing et al. [31]	Only text	SVM-TS RFC tanh-RNNs (single hidden layer) LSTM-1 (one hidden layer) GRU-1 (one-layer) GRU-2 (extra GRU hidden layer)	Twitter 498 rumors and 494 non-rumors, Sina Weibo 2,313 rumors and 2,351 non-rumors.
Muhammad et al. [32]	Only text	K-nearest neighbors (KNN), Decision tree (DT), Random forest (RF), Logistic regression (LR), Naïve Bayes (NB) LSTM, CNN, RNN, LSTM-CNN. BiLSTM-CNN	Pheme rumor dataset (5800 tweets)
Sansiri et al. [37]	Only text	LSTM CNN	20,015 news, (11,941 fake 8,074 real news)

(Continued)

Table 2: Continued

Study	Used features	Experimented models	Dataset
Verma et al. [34]	Only text	GRU RRN + LSTM	50,000 real news dataset
Kaliyar et al. [35]	Only text	CNN	Kaggle dataset FakeNews Dataset
Kaliyar et al. [36]	Text the behavior of the source speaker.	CNN BSLTM MLP	LIAR Dataset
Sansiri et al. [37]	Content-based author-based features	DTC RFC SVM Ensemble RNN GRU-2 CNN	Real-world dataset (nearly 615000 tweets from 284000 users)
Pavithra et al. [38]	Textural propagation features	CNN RNN RvNN	Pheme-RNR (1971 rumor + 3819 non-rumor)
Jing et al. [39]	Text propagation-based features	DTR, DTC RFC, Different variations of SVM, GRU-RNN. BU-RvRNN TD-RvRNN	Twitter15 (1,381 propagation trees) Twitter16 (1,181 propagation trees)
Yichun et al. [40]	Post-based, User-based	DLSTM DGRU DSRNN	Sina Weibo, (2313 rumors, 2351 non-rumors) Twitter dataset 111 events (60 rumors and 51 non-rumors) and 192,350 posts.
Lin et al. [41]	Text user-based content-based signal features	DTR DTC SVM GRUs hierarchical RNN BSLTMs	Twitter (992 events) Sina Weibo (4664 events)

2.4 Arabic Rumor Detection

The Arabic language has a complex structure that imposes challenges in addition to the lack of datasets. Thus, the researches on rumor detection in Arabic social media are few and require more attention and effort to achieve optimal results. The studies that are focused on Arabic rumor detection are summarized in Tab. 3.

Table 3: The researches that are concerning Arabic rumors detection

Reference	Year	Objective	Method
[42]	2016	- Determining the attributes of Arabic rumor (fake information) patterns. Showing Arabic rumor identification in Twitter social.	Using natural language processing and machine learning Using TF-IDF (Term frequency-inverse document frequency) for determining the weights for the terms.
[43]	2018	- Detecting truth in the Arabic tweets - Preview the related work of credibility check for Arabic.	- Introducing main procedures of the public model of Truth Detection for Arabic Tweets
[44]	2018	- Detecting the trustiness of the information that is prevalent through social media.	- Using classical machine learning algorithms - Using Cluster features and Cluster-Post features that were collected from social media
[45]	2018	- Credibility checking model for Arabic news	- Content parsing, features extraction, content verification, users' comments, polarity evaluation, and credibility classification.
[46]	2019	- Detecting rumors in Arabic social media	- Using extracted features from the user and the content and analyzing to identify their importance. - Utilizing two different learning models, semi-supervised learning and unsupervised learning using expectation–maximization

3 Proposed Architecture

The proposed methodology investigates the most famous state of the arts deep learning algorithms for Arabic text classification; like deep learning techniques, they have the advantage of their ability to capture semantic features from textual data [47] automatically. The four different combinations of deep neural networks, namely, CNN, LSTM, BiLSTM, CNN+LSTM, and CNN + BiLSTM classification, have been performed. The proposed methodology is shown in Fig. 1.

3.1 Datasets

The first dataset consists of news and tweets that were manually collected and annotated by rumor/non-rumor. The actual dataset was collected from the Arabic news portals such as; Youm7, Akhbarelyom, and Ahram. This fake news was announced to make people aware that news is not absolute and fake. This effort is the responsibility of the information and decision support center, the Egyptian cabinet.

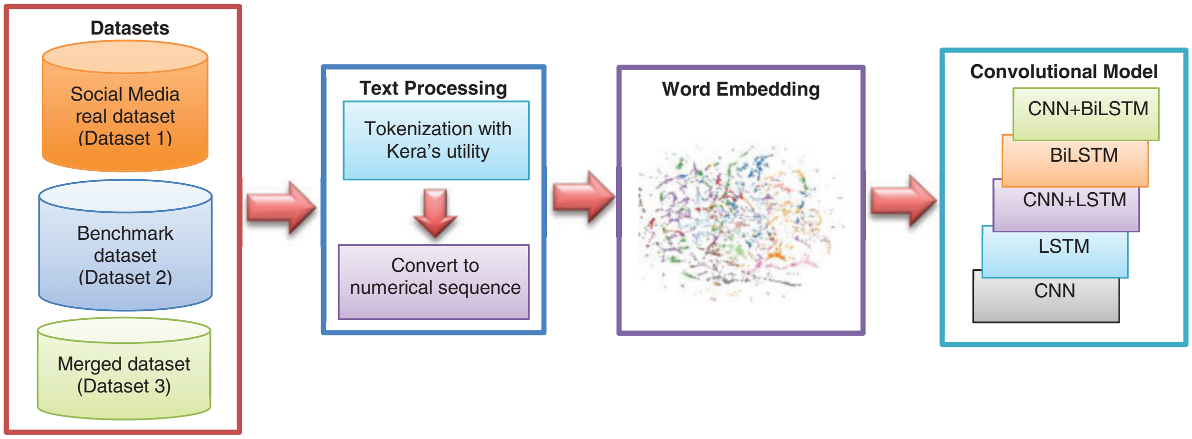

Datasets
Social Media real dataset (Dataset 1)
Benchmark dataset (Dataset 2)
Merged dataset (Dataset 3)
Text Processing
Tokenization with Kera's utility
Convert to numerical sequence
Word Embedding
Convolutional Model
CNN+BiLSTM
BiLSTM
CNN+LSTM
LSTM
CNN

Figure 1: System architecture

The second dataset is a benchmark dataset published in [48]. Then, the two datasets are merged into one large combined dataset to test deep learning performance using a larger dataset. The details of each dataset are shown in the Tab. 4.

Table 4: The description of the datasets

Dataset	Number of tweets	Class label 0 (non-rumor)	Class label 1 (rumor)	No unique tokens	The average length of tokens	Max text length
Real dataset (Dataset 1)	1980	801	1178	8881	14.1	46
Benchmark dataset (Dataset 2)	2578	1218	1364	12919	18.4	88
Merged dataset (Dataset 3)	4561	2019	2542	18483	16.3	88

Tab. 5 shows samples of that real dataset collected from Arabic news websites.

3.2 *Preprocessing*

Preprocessing the text before it is fed into the classifier is very important and impacts the overall performance of the classification model. In this step, the text is cleaned using filters to remove punctuations and all non-Unicode characters. Afterward, stop words are removed, then sentences are tokenized, and tokens are stemmed. The resulting sentences are then encoded as numerical sequences, the number of unique tokens and the maximum sentences' length is calculated. This maximum length is used to pad all sentences to be of the same size, equal to the maximum length. Labels are then encoded using one-hot-encoding.

Table 5: The samples of the real dataset

Arabic fake news	Arabic real news
رفع سعر رغيف الخبز المدعم بعد تطبيق منظومة بيع القمح النقدي الجديدة بدايةً من يوليو القادم	وكيلة الأمين العام للأمم المتحدة تشيد بجهود مصر فى تضمين احتياجات المرأة لمواجهة كورونا
تراجع وزارة التعليم العالي عن قرار إتمام امتحانات الفصل الدراسي الثاني في الجامعات للفرق النهائية	برلمانى: قرارات الحكومة الأخيرة ستسهم فى تحسن الأداء بمواجهة كورونا
عدم تسليم طلاب الثانوية العامة "بوكليت" المواد غير المضافة للمجموع	برلمانى: مصر تطبق خطة احترافية فى مواجهة كورونا.. وحملات التشكيك "بائسة"
بيع بقالي التموين السلع التموينية بأسعار مخالفة للأسعار المقررة من قبل الوزارة	رئيس "خطة البرلمان": علينا استغلال تجربة كورونا لإصلاح الجهاز الإدارى
طرح الكمامات القماشية على البطاقات التموينية بسعر ٤٠ جنيهاً بدايةً من يوليو المقبل	برلماني: مصر على أبواب ذروة كورونا والوعي والالتزام ينقذان الوطن
إجبار المواطنين على شراء المطهرات على بطاقات التموين	المؤتمر: جيش مصر الأبيض أفشل مؤمرات الإرهابية فى مواجهتهم مع كورونا

3.3 *Word Embedding*

Recently, word embeddings proved to outperform traditional text representation techniques. It represents each word as a real-valued vector in a dimensional space while preserving the semantic relationship between words. As vectors of words with similar meanings are placed close to each other. Word embeddings can be learned from the text to fit a deep neural model on text data. For our work, the Tensorflow Keras embedding layer was used. It takes, as an input, the numerically encoded text. It is implemented as the first hidden layer of the deep neural network where the word embeddings will be learned while training the network. The embedding layer stores a lookup table to map the words represented by numeric indexes to their dense vector representations.

3.4 *Proposed Models*

Our system explores the usage of three deep neural networks, namely, CNN, LSTM, and BiLSTM, and two combinations between CNN+LSTM and CNN+ BiLSTM as illustrated in Fig. 2.

CNN model consists of one conventional layer which learns to extract features from sequences represented using a word embedding and derive meaningful and useful sub-structures for the overall prediction task. It is implemented with 64 filters (parallel fields for processing words) with a linear ('relu') activation function. The second layer is a pooling layer to reduce the output of the convolutional layer by half. The 2D output from the CNN part of the model is flattened to one long 2D vector to represent the 'features' extracted by the CNN. Finally, two dense layers are used to scale, rotate and transform the vector by multiplying Matrix and vector.

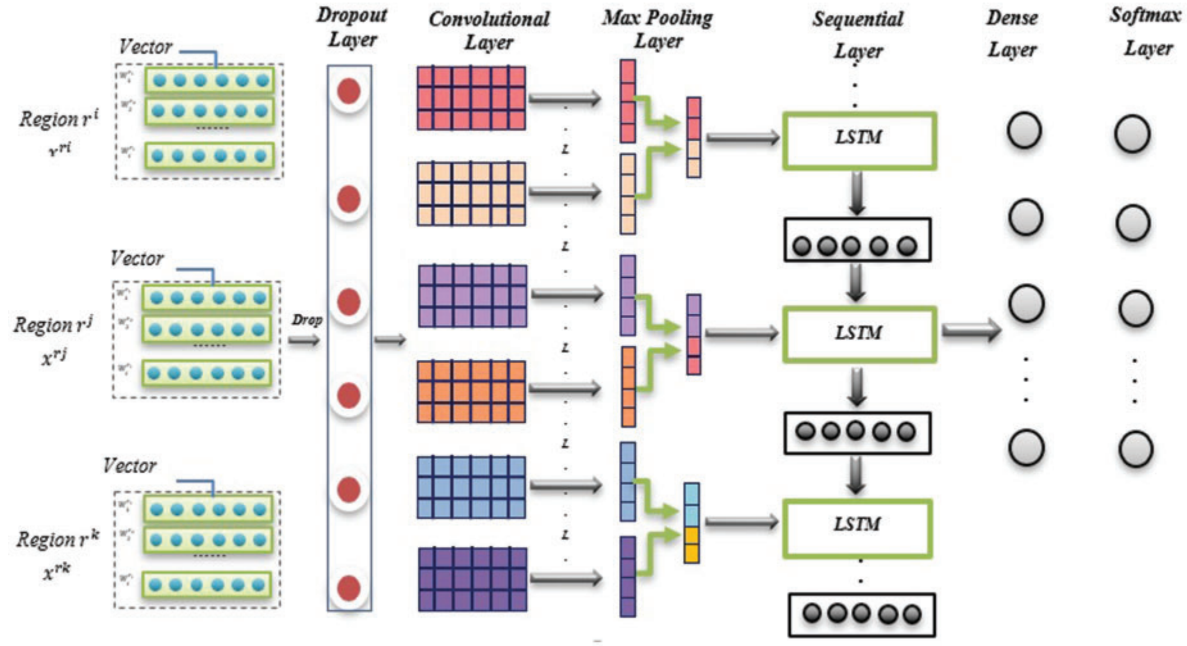


Figure 2: Deep learning model

The output of the word embedding layer is fed into one LSTM layer with 128 memory units. The output of the LSTM is fed into a dense layer of size = 64. This is used to enhance the complexity of the LSTM's output threshold. The activation function is natural for binary classification. It is a binary-class classification problem; binary cross-entropy is used as the loss function.

The third model combines both CNN with LSTM, where two conventional layers are added with max-pooling and dropout layers. The conventional layers act as feature extractors for the LSTMs on input data. The CNN layer uses the output of the word embeddings layer. Afterward, the pooling layer reduces the features extracted by the CNN layer. A dropout layer is added to help to prevent neural from overfitting. The LSTM layer with a hidden size of 128 is added. We use one LSTM layer with a state output of size = 128. Note, as per default return sequence is False, we only get one output, i.e., of the last state of the LSTM. The output of LSTM is connected with a dense layer of size = 64 to produce the final class label by calculating the probability of the LSTM output. The softmax activation function is used to generate the final classification.

The BiLSTM model uses a combination of recurrent and convolutional cells for learning. The output from the word embeddings layers is fed into a bi-directional LSTM. Afterward, dense layers are used to find the most suitable class based on probability.

BiLSTM-CNN model architecture uses a combination of convolutional and recurrent neurons for learning. As input, the output of the embeddings layers is fed into two-level conventional layers for feature learning for the BiLSTM layer. The features extracted by the CNN layers are max-pooled and concatenated. The fully connected dense layer predicts the probability of each class label.

For Training the deep learning models, Adam optimizer with 0.01 learning rate, weight decay of 0.0005, and 128 batch size. A dropout value of 0.5 is used to avoid overfitting and speed up the learning. The output layer uses a softmax activation function.

The experiments used python programming Tensorflow and Keras libraries for machine learning and deep learning models. A windows 10–based machine with core i7 and 16 GB RAM was used.

4 Results and Discussion

Two experiments have been performed on three different datasets. The first experiment utilizes the proposed deep learning algorithms. The second experiment utilizes machine-learning algorithms using n-gram feature extraction and compares their results with deep learning algorithms.

Experiments included two phases; first, the most famous machine learning algorithms have been applied for classification with different n-grams. Machine learning techniques were evaluated using accuracy, f1-measure, and AUC (Area Under Curve) measures. The second phase of experiments included applying deep learning models for classification. Deep learning algorithms were first trained using simple data spilled with 80% training and 20% testing. Then same algorithms were trained using 5-fold cross-validation [49].

4.1 Machine Learning Experiments

The experiments are conducted using many machine learning algorithms, including Linear SVC, SVC, multinomialNB, bernoulliNB, stochastic gradient descent (SGD), decision tree, random forest, and k-neighbors. Each algorithm is evaluated using accuracy, F-score, and area under the curve (AUC). The results of the first dataset experiment are shown in Tab. 6. The table shows that the SGD classifier gives the best results. The figure shows that SVC, decision tree, and random forest give lower performance than other algorithms.

Table 6: The results of the first dataset experiment

Algorithm	Accuracy	F1_score	AUC
Linear SVC	0.859	0.857	0.845
SVC	0.596	0.445	0.5
MultinomialNB	0.823	0.814	0.788
BernoulliNB	0.821	0.811	0.786
SGD classifier	0.861	0.861	0.853
Decision tree classifier	0.641	0.56	0.563
Random forest classifier	0.596	0.445	0.5
KNeighbors classifier	0.806	0.796	0.77

The results of the second dataset experiment are shown in Tab. 7. Each algorithm is evaluated using accuracy, F-score, and area under the curve (AUC). Tab. 7 shows that the LinerarSVC classifier gives the best results. The table shows that SVC, decision tree, and random forest give lower performance than other algorithms.

Table 7: The results of the second dataset experiment

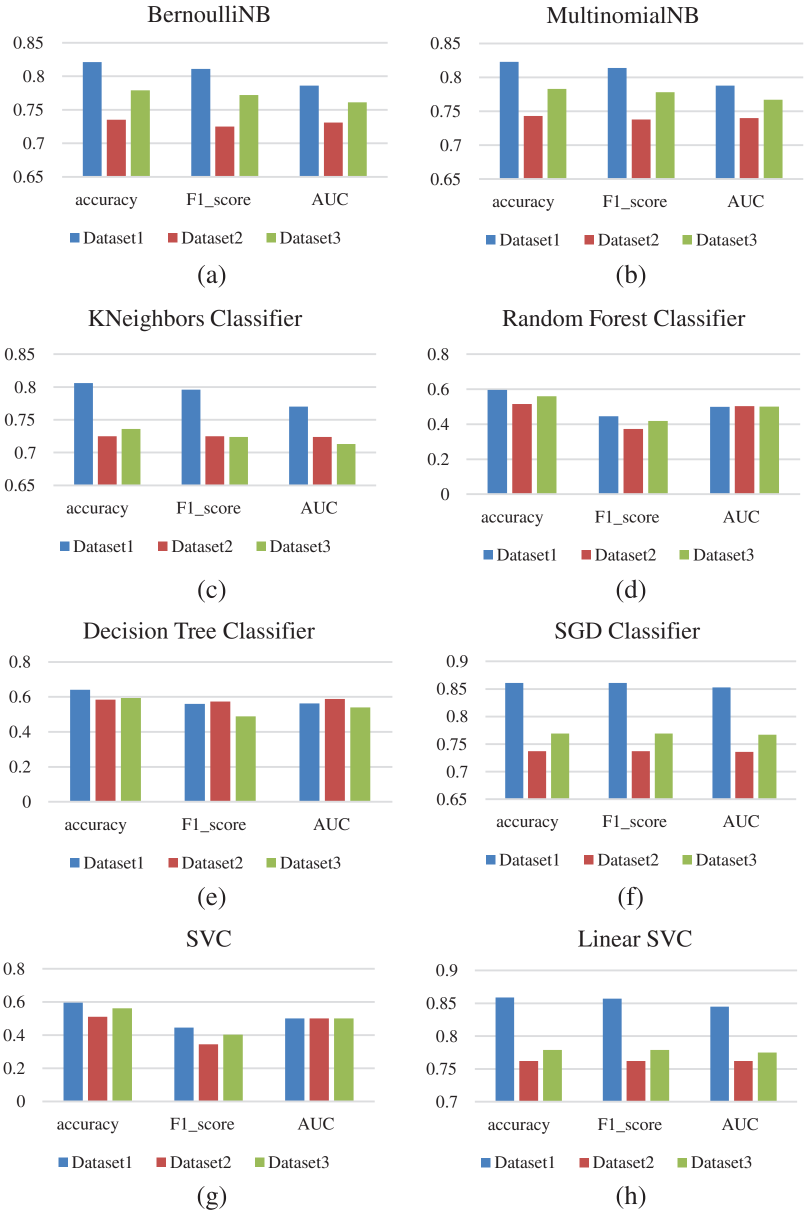
Algorithm	Accuracy	F1_score	AUC
Linear SVC	0.762	0.762	0.762
SVC	0.511	0.345	0.5
MultinomialNB	0.743	0.738	0.74
BernoulliNB	0.735	0.725	0.731
SGD classifier	0.737	0.737	0.736
Decision tree classifier	0.584	0.573	0.588
Random forest classifier	0.515	0.373	0.504
KNeighbors classifier	0.725	0.725	0.724

The results of the third dataset experiment are shown in Tab. 8. Each algorithm is evaluated using accuracy, F-score, and area under the curve (AUC). The table shows that the LinerarSVC classifier gives the best F-score and AUC while MultinomialNB gives the best accuracy. The figure shows that SVC, decision tree, and random forest give lower performance than other algorithms.

Table 8: The results of the third dataset experiment

Algorithm	Accuracy	F1_score	AUC
Linear SVC	0.779	0.779	0.775
SVC	0.561	0.403	0.5
MultinomialNB	0.783	0.778	0.767
BernoulliNB	0.779	0.772	0.761
SGD classifier	0.769	0.769	0.767
Decision tree classifier	0.593	0.489	0.54
Random forest classifier	0.56	0.419	0.501

The methods SVC, decision tree, and random forest can be concluded that is not suitable for this problem. The following graphs depicted in Figs. 3a–3h show the performance of each ML algorithm applied to each dataset. The BernoulliNB method shows its performance with the first dataset, found in Fig. 3a. Fig. 3b shows that MultinomialNB gives its lower performance for the third dataset. Fig. 3c shows that k neighbors give their best performance for the first dataset. Fig. 3d shows that random forest gives its best performance for the first dataset. Fig. 3e shows that the decision tree performs best for the first dataset. Fig. 3f shows that the SGD classifier gives an almost equivalent performance for all datasets. Fig. 3g shows that SVC gives its best performance for the first dataset. Fig. 3h shows that Linear SVC gives almost equivalent performance for all datasets. The first dataset is the manual collected and annotated data, which is the real-life data. Therefore, the machine learning algorithms give excellent performance for real-life data.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3: The performance of each ML algorithms with each dataset

4.2 Deep Learning Algorithms

Many deep learning algorithms are conducted; CNN, LSTM, CNN + LSTM, BiLSTM, and CNN + BiLSTM. The evaluation metrics of accuracy, loss, and AUC are used.

Tab. 9 shows the performance of each algorithm applied to the first dataset. Tab. 9 shows that BiLSTM gives the slightest loss, best accuracy, and best AUC. Thus, BiLSTM gives good performance with reasonable loss compared to other algorithms, which are close to each other in performance.

Table 9: The performance of each algorithm applied on the first dataset

Deep learning model	Accuracy	Loss	AUC
CNN	0.780303	0.582185	0.839837
LSTM	0.838384	1.387679	0.887626
CNN + LSTM	0.825758	0.837274	0.886626
BiLSTM	0.848283	0.583095	0.902887
CNN + BiLSTM	0.8283	0.909429	0.886772

Tab. 10 shows the performance of each algorithm on the second dataset. Results on the second dataset also show that BiLSTM gives the least amount of loss, the best accuracy, and the best. Additionally, CNN gives a bad performance with a significant loss and the lowest accuracy. Other algorithms give almost similar performances.

Table 10: The performance of each algorithm applied on the second dataset

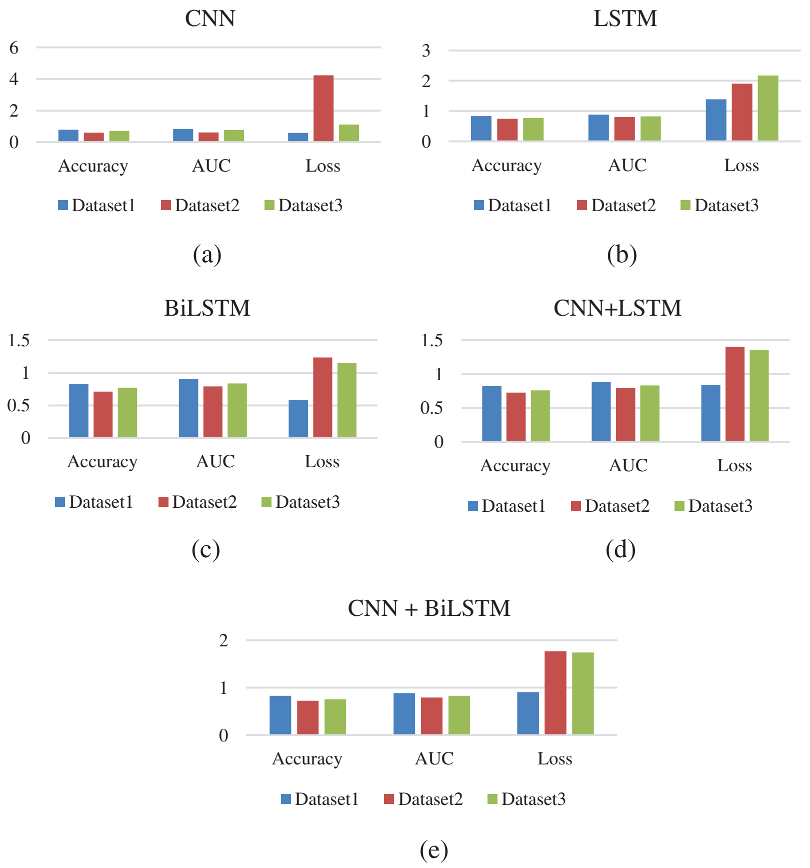
Deep learning model	Accuracy	Loss	AUC
CNN	0.597679	4.226126	0.61282
LSTM	0.709865	1.902138	0.792778
CNN + LSTM	0.727273	1.401511	0.792778
BiLSTM	0.742747	1.236142	0.803262
CNN + BiLSTM	0.7273	1.768434	0.791577

Tab. 11 shows the performance of each algorithm on the third dataset. Tab. 11 shows that CNN gives the least amount of loss. BiLSTM gives the best accuracy and the best AUC. Tab. 11 shows that LSTM and CNN + BiLSTM give a significant loss while accuracies and AUC are almost similar to other algorithms.

Table 11: The performance of each algorithm applied on the third dataset

Deep learning model	Accuracy	Loss	AUC
CNN	0.710843	1.118824	0.765603
LSTM	0.765608	2.172314	0.829969
CNN + LSTM	0.760131	1.359961	0.831611
BiLSTM	0.773275	1.151952	0.837928
CNN + BiLSTM	0.7558	1.741797	0.829785

The following graphs depicted in Figs. 4a–4h show the performance of each deep learning algorithm with each dataset. The BiLSTM method is more suitable for the first and second datasets because it significantly loses the third dataset, as shown in Fig. 4a. Fig. 4b shows that CNN is not suitable for this problem as it gives low performance for all datasets and a considerable amount of loss in the third dataset. Fig. 4c shows that CNN + BiLSTM is more suitable for datasets one and second as it significantly loses dataset three. Fig. 4d shows that CNN+LSTM is more suitable for the first and second datasets as it gives a significant amount of loss in the third dataset. Fig. 4e shows that LSTM provides accepted performance for the first and second datasets but encounters a significant loss for the third dataset. Therefore, deep learning algorithms give better performance when combining with real-life data.



(a) (b)

(c) (d)

(e)

Figure 4: The performance of each deep learning algorithm with each dataset

4.3 Cross-Validation

To verify the experiments done with deep learning algorithms, five-fold cross-validation on the three datasets have experimented. The results of each dataset are shown in Tabs. 12–14. Results show that BiLSTM and BiLSTM + CNN give the highest accuracy and most negligible loss for all three datasets. On the other hand, CNN achieved the worst performance among all experimented models.

Table 12: The five-fold cross-validation of the first dataset

Deep learning model	Accuracy	Loss
CNN	58.27	4.54
LSTM	82.06	1.13
CNN + LSTM	81.86	1.01
BiLSTM	83.92	0.86
CNN + BiLSTM	83.88	1.08

Table 13: The five-fold cross-validation of the second dataset

Deep learning model	Accuracy	Loss
CNN	58.43	4.25
LSTM	70.02	2.51
CNN + LSTM	70.19	2.09
BiLSTM	70.83	2.06
CNN + BiLSTM	71.72	2.01

Table 14: The five-fold cross-validation of the third dataset

Deep learning model	Accuracy	Loss
CNN	57.42	5.94
LSTM	73.34	2.59
CNN + LSTM	73.03	2.13
BiLSTM	74.98	1.97
CNN + BiLSTM	73. 87	2.02

5 Conclusions and Future Works

This paper aims at investigating machine learning and deep learning models for content-based Arabic fake news classification. A series of experiments were conducted to evaluate the task-specific deep learning models. Three datasets were used in the experiments to assess the most well-known models in the literature. Our findings indicate that machine learning and deep learning approaches can identify fake news using text-based linguistic features. There was no single model that performed optimally across all datasets in terms of machine learning algorithms. On the other hand, our results show that the BiLSTM model achieves the highest accuracy among all models assessed across all datasets.

We intend to thoroughly examine the existing architectures combining various layers as part of our future work. Furthermore, examine the effect of various pre-trained word embeddings on the performance of deep learning models.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] M. Vohra and M. Kakkar, "Detection of rumor in social media," in *Proc. 8th Int. Conf. on Cloud Computing, Data Science & Engineering*, Noida, India, pp. 485–490, 2018.

[2] F. Pierri and S. Ceri, "False news on social media: A data-driven survey," *ACM SIGMOD Record*, vol. 48, no. 2, pp. 18–27, 2019.

[3] S. Kumar and N. Shah, "False information on web and social media: A survey," *ArXiv*, vol. abs/1804.08559, 2018.

[4] S. Zannettou, M. Sirivianos, J. Blackburn and N. ourtellis, "The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans," *Journal of Data and Information Quality (JDIQ)*, vol. 11, no. 10, pp. 1–37, 2019.

[5] C. Tandoc, W. Lim and R. Ling, "Defining "fake news" a typology of scholarly definitions," *Digital Journalism*, vol. 6, no. 2, pp. 137–153, 2018.

[6] Q. Wang, X. Yang and W. Xi, "Effects of group arguments on rumor belief and t]transmission in online communities: An information cascade and group polarization perspective," *Information & Management*, vol. 55, no. 4, pp. 441–449, 2018.

[7] V. Agarwal, H. Sultana, S. Malhotra and A. Sarkar, "Analysis of classifiers for fake news detection," *Procedia Computer Science*, vol. 165, no. 1, pp. 377–383, 2019.

[8] Y. Peng, Z. Zhang, X. Wang, L. Yang and L. Lu, "Chapter 5-text mining and deep learning for disease classification," in *The Elsevier and MICCAI Society Book Series, Handbook of Medical Image Computing and Computer Assisted Intervention*, Academic Press, 2020.

[9] R. Kaliyar, A. Goswami, P. Narang and S. Sinha, "FNDNet a deep convolutional neural network for fake news detection," *Cognitive Systems Research*, vol. 61, no. 1, pp. 32–44, 2020.

[10] A. Elnagar, R. Al-Debsi and O. Einea, "Arabic text classification using deep learning models," *Information Processing and Management*, vol. 57, no. 1, pp. 102–121, 2020.

[11] V. Pérez-Rosas, B. Kleinberg, A. Lefevre and R. Mihalcea, "Automatic detection of fake news," in *proc. 27th Int. Conf. on Computational Linguistics*, Santa Fe, New Mexico, USA, pp. 3391–3401, 2018.

[12] Z. Jin, J. Cao, Y. Zhang, J. Zhou and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 598–608, 2017.

[13] M. Ahsan, M. Kumari and T. Sharma, "Rumors detection, verification and controlling mechanisms in online social networks: A survey," *Online Social Networks and Media*, vol. 14, no. 1, pp. 1–12, 2019.

[14] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences*, vol. 497, no. 1, pp. 38–55, 2019.

[15] 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.

[16] S. Sabbeh,"Performance evaluation of different data mining techniques for social media news credibility assessment," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, pp. 245–256, 2019.

[17] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra *et al.*, "Elstream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning," *IEEE Access*, vol. 9, pp. 66408–66419, 2021.

[18] J. Reis, A. Correia, F. Murai, A. Veloso and F. Benevenuto, "Supervised learning for fake news detection," *IEEE Intelligent Systems*, vol. 34, no. 2, pp. 76–81, 2019.

[19] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," in *IEEE First Ukraine Conf. on Electrical and Computer Engineering (UKRCON)*, Kyiv, Ukraine, pp. 900–903, 2017.

[20] M. Benjamin, C. Ciro and M. Filippo, “Social spam detection,” in *5th Int. Workshop on Adversarial Information Retrieval on the Web*, ACM, Madrid, Spain, pp. 41–48, 2009.

[21] R. Deepa and N. Radha, “Supervised learning approach for spam classification analysis using data mining tools,” *(IJCSE) International Journal on Computer Science and Engineering*, vol. 2, no. 9, pp. 2783–2789, 2010.

[22] K. Fouad, T. Elsheshtawy and M. Dawood, “Intelligent approach for large-scale data mining,” *International Journal of Sociotechnology and Knowledge Development*, vol. 13, no. 2, pp. 119–152, 2021.

[23] Z. Jin, J. Cao, Y. Zhang and J. Luo, “News verification by exploiting conflicting social viewpoints in microblogs,” in *Thirtieth AAAI Conf. on Artificial Intelligence*, Phoenix, Arizona, USA, pp. 2972–2978, 2016.

[24] L. Siwei, X. Liheng, L. Kang and Z. Jun, “Recurrent convolutional neural networks for text classification,” in *Twenty-Ninth AAAI Conf. on Artificial Intelligence*, Austin, Texas, USA, pp. 2267–2273, 2015.

[25] A. Oscar, C. Ignacio, J. Fernando and A. Carlos, “Enhancing deep learning sentiment analysis with ensemble techniques in social applications,” *Expert Systems with Applications*, vol. 77, pp. 246, 2017.

[26] A. Elnagar, R. Al-Debsi and O. Einea, “Arabic text classification using deep learning models,” *Information Processing and Management*, vol. 57, no. 1, pp. 102121, 2020.

[27] M. Heikal, M. Torki and N. El-Makky, “Sentiment analysis of arabic tweets using deep learning,” in *4th Int. Conf. on Arabic Computational Linguistics (ACLing 2018)*, Dubai, UAE, pp. 114–122, 2018.

[28] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti and M. Akbar, “Fake news detection using deep learning models: A novel approach,” *Transactions on Emerging Telecommunication Technologies*, vol. 31, no. 2, pp. e3767, 2020.

[29] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi *et al.*, “Fake news stance detection using deep learning srchitecture (CNN-lSTM),” *IEEE Access*, vol. 8, pp. 156695–156706, 2020.

[30] S. Girgis, E. Amer and M. Gadallah, “Deep learning algorithms for detecting fake news in online text,” in *13th Int. Conf. on Computer Engineering and Systems (ICCES)*, Cairo, Egypt, pp. 93–97, 2018.

[31] M. Jing, G. Wei, M. Prasenjit, K. Sejeong, J. Jim *et al.*, “Detecting rumors from microblogs with recurrent neural networks,” in *Twenty-Fifth Int. Joint Conf. on Artificial Intelligence IJCAI-16*, New York, USA, pp. 3818–3824, 2016.

[32] A. Muhammad, H. Ammara, H. Anam, K. Adil, A. Rehman *et al.*, “Exploring deep neural networks for rumor detection,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 4315–4333, 2019.

[33] R. AlvaroIbrain and L. Lara, “Fake news detection using deep learning,” *Journal of Information Process Systems*, vol. 15, no. 5, pp. 1119–1130, 2019.

[34] A. Verma, V. Mittal and S. Dawn, “FIND: Fake information and news detections using deep learning,” in *Twelfth Int. Conf. on Contemporary Computing (IC3)*, Noida, India, pp. 1–7, 2019.

[35] 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.

[36] R. Kaliyar, A. Goswami and P. Narang, “FakeBERT: Fake news detection in social media with a BERT-based deep learning approach,” *Multimedia Tools and Application*, vol. 80, no. 8, pp. 11765–11788, 2021.

[37] T. Sansiri and H. Kien “Attention based neural architecture for rumor detection with author context awareness,” in *Thirteenth Int. Conf. on Digital Information Management (ICDIM)*, Luton, UK, pp. 82–87, 2018.

[38] C. Pavithra and J. Shibily, “Deep learning approach for rumour detection in twitter: A comparative analysis,” in *Int. Conf. on Systems, Energy & Environment (ICSEE)*, GCE Kannur, Kerala, 2019.

[39] M. Jing, G. Wei and W. Kam-Fai, “Rumor detection on twitter with tree-structured recursive neural networks,” in *56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, Melbourne, Australia, pp. 1980–1989, 2018.

[40] X. Yichun, W. Chen, D. Zhiping, S. Shuifa and D. Fangmin, “Deep recurrent neural network and data filtering for rumor detection on sina weibo,” *Symmetry*, vol. 11, no. 11, pp. 1408, 2019.

[41] X. Lin, X. Liao, T. Xu, W. Pianand and K. Wong, “Rumor detection with hierarchical recurrent convolutional neural network,” in *8th CCF Int. Conf., NLPCC*, Dunhuang, China, pp. 338–348, 2019.

[42] A. Yahya, "Arabic rumours identification by measuring the credibility of arabic tweet content," *International Journal of Knowledge Society Research (IJKSR)*, vol. 7, no. 2, pp. 72–83, 2016.

[43] R. Mouty and A. Gazdar, "Survey on steps of truth detection on arabic tweets," in *21st Saudi Computer Society National Computer Conf. (NCC)*, Riyadh, Saudi Arabia, pp. 1–6, 2018.

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

[45] S. Sabbeh and S. Baatwah, "Arabic news credibility on twitter: An enhanced model using hybrid features," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 8, pp. 2327–2338, 2018.

[46] S. Alzanin and A. Azmi, "Rumor detection in arabic tweets using semi-supervised and unsupervised expectation–maximization," *Knowledge-Based Systems*, vol. 185, pp. 104945, 2019.

[47] Q. Liu, Y. Huang, Y. Gao, X. Wei, Y. Tian *et al.,* "Task-oriented word embedding for text classification," in *27th Int. Conf. on Computational Linguistics*, Santa Fe, New-Mexico, USA, pp. 2023–2032, 2018.

[48] R. Francisco, R. Paolo, C. Anis, Z. Wajdi, G. Bilal *et al.*, "Overview of the track on author profiling and deception detection in arabic," in *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019)*, Kolkata, India, pp. 70–83, 2019.

[49] K. Fouad and D. El-Bably, "Intelligent approach for large-scale data mining," *International Journal of Computer Applications in Technology*, vol. 63, no. 1–2, pp. 93–113, 2020.