Deep Learning Based Rumor Detection for Arabic Micro-Text

Shada Alharbi, Khaled Alyoubi, Fahd Alotaibi
salharbi0884@stu.kau.edu.sa
Faculty of Computer Science and Information Technology, Jeddah, Saudi Arabia

Abstract
Nowadays microblogs have become the most popular platforms to obtain and spread information. Twitter is one of the most used platforms to share everyday life event. However, rumors and misinformation on Arabic social media platforms has become pervasive which can create inestimable harm to society. Therefore, it is imperative to tackle and study this issue to distinguish the verified information from the unverified ones. There is an increasing interest in rumor detection on microblogs recently, however, it is mostly applied on English language while the work on Arabic language is still ongoing research topic and need more efforts. In this paper, we propose a combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to detect rumors on Twitter dataset. Various experiments were conducted to choose the best hyper-parameters tuning to achieve the best results. Moreover, different neural network models are used to evaluate performance and compare results. Experiments show that the CNN-LSTM model achieved the best accuracy 0.95 and an F1-score of 0.94 which outperform the state-of-the-art methods.

Key words:
Rumor Detection, Natural Language Processing, Convolutional Neural Network, Long Short-Term Memory

1. Introduction

Due to the increasing use of internet around the world, it is become very easy to spread rumors and fake news. According to [1] an increasing amount of people rely on information shared via social media platforms. This includes Facebook, YouTube, Twitter etc. However, there is no restriction while posting any news on these platforms. Thus, some people may spread fake news against certain individuals or organizations. Twitter has been considered as one of the most widely used social networking platform for sharing and spreading news in the Arab world [2]. The ease nature of Twitter to share and spread news with others makes it a challenging environment to study and analyze the behavior of deceptive and unverified information and how to eliminate and detect them. In recent years, the use of Arabic language in social media has been increased. Despite the number of studies published in different languages about detecting rumors and misinformation, there is a lack of Arabic studies being conducted in the same language compared to other languages especially English.

A rumor can be defined as the information whose truth value is unverifiable [3]. Likewise, another paper [4] define rumor as unverified information that may not be untruthful in some cases. In contrast, Seo et al. [5] consider rumor as a false information anyway. On Twitter, a rumor is a set of tweets contain the same unverified statement, which propagate through the network. Thus, a rumor can be spread easily on the web especially microblogs platforms, resulting in widespread real-world impact. Rumor detection is one of the most challenging tasks in text classification which is the process of assigning categories or classes to a certain text based in its content. It is one of the essential tasks in Natural Language Processing (NLP) which has attracted many researchers in this filed recently.

Machine learning which is part of artificial intelligence that automate the learn process of any system in order to perform different actions [6]. Machine learning classifiers includes supervised, unsupervised, and reinforcement algorithms can be used for different purposes such as rumor and fake news detection.

However, due to the emergence of deep learning, it has been widely considered that it can improve text classification tasks and in rumor detection field compared to traditional methods. DL models such as convolutional neural networks (CNN) and recurrent neural networks (RNN) have shown remarkable results in text classification tasks. Therefore, our work in this paper follows this approach to develop a DL model for detecting rumors on Arabic language. A hybrid DL model is proposed to examine rumor detection on social media platform. The proposed approach is a combination of CNN and LSTM architectures which called CNN-LSTM. The experimental results showed the effectiveness of our hybrid model compared to the other existing works.

The major contribution of this paper is as follows:

- Review the existing research on rumor detection on social media.
- Use deep learning methods to classify text as rumor or non-rumor.
- Examine the use of word embedding over the traditional features and compare results.
- Compare the efficiency of our proposed model with other baseline methods.

The remainder of the paper is organized as follows: in section 2 we demonstrate related work. The used dataset is described in section 3. In section 4 and 5, we present the proposed methodology and analyze the results. Finally, section 6 concludes the study and gives some recommendation for the future work.

Manuscript received November 5, 2021
Manuscript revised November 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.11.10

2. Motivation

Detecting rumors is a critical task in reducing the dissemination of misleading knowledge within social networking sites. The work on English language is active and has a lot of contributions unlike the Arabic language. The field of rumor detection on Arabic language is still in an early stage compared to the other languages. However, Arabic language considered as one of the major languages in the world. Arabic text classification is an active research topic these days. Though, it is a complicated classification problem and has many challenges that need to be tackled. However, there are several directions for enhancing the Arabic rumor detection field. In general, studying rumor on social media requires several steps to classify a certain tweet either rumor or non-rumor. Figure 1 represent the general workflow of the proposed system. We note from our literature study that the most used methods for detecting rumor on Arabic language is machine learning-based algorithms which is less effective compared to deep learning-based algorithms. Furthermore, there are many challenges related to Arabic language in term of morphology, dialects and linguistic problems in general. The complexity and richness of the Arabic morphology makes it difficult to interpret any word to its actual meaning. Moreover, the lack of corpora and lexicon for rumor detection created additional challenges especially for the user-generated content due to its noisy nature. In this study, we focus on detecting rumor on Arabic tweets.

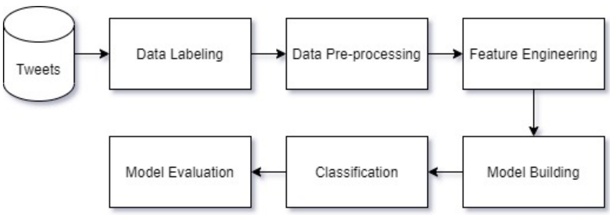

Fig 1 Block diagram of the proposed system

3. Related Work

Rumor detection is one of the more crucial research topics in recent years. In this section, we summarize the work done in the field of rumor detection using both deep learning and machine learning techniques. Floos [7] focus on detecting rumors for Arabic tweets. the author gathered two types of tweets Rumors and News collected from different Twitter accounts and calculate TF-IDF for each term. Then, they compare the scalars for the tweets vector and the news vector using dot product. The largest scalar will be classified as a news tweet. A study published on 2019 [8] that detect Arabic rumors on Twitter using content-based and user-based features. They used semi-supervised expectation-maximization (E-M) model and achieved an f1 score of 80%. Thakur et al. [9] proposed a framework using machine learning algorithm to detect rumors. The framework focuses on filtering the linguistic properties of the text to analyze the data if it is rumor or not.

As for deep learning methods, Kaliyar et al. [10] proposed a hybrid model using CNN and LSTM layers in addition to three dense layers. Also, they created a dataset (FN-COV) that contains news article about COVID-19 pandemic. Their experiments were conducted on another dataset called PHEME dataset [1]. The authors observed that their model performed better on FN-COV dataset with accuracy of 98.62%. their model achieved remarkable results because it can capture both temporal semantics and phrase-level representations. Chen et al. [11] have proposed a hybrid model called XGA which is developed mainly using XLNet and Bidirectional Gated Recurrent Unit (Bi-GRU) with Attention mechanism. The model is built to detect Cantonese rumors on Twitter using semantic and sentiment features. However, the semantic information is discovered automatically using the attention mechanism. The experiment results show that their model outperform the other models used to detect Cantonese rumors. For the graph based approach, Zhang et al. [12] proposed an aggregation graph neural network architecture on rumor detection to reduce the computational complexity. They observe that the aggregation operation can capture different characteristics of distinct rumors. Bian et al. [13] proposed a bi-directional graph convolutional neural network that study propagation of rumors. Thus, the model depends on propagation and dispersion features by working in top-down and bottom-up rumor propagation.

Similar to Kaliyar et al.[10] , another approach proposed by Shi et al. [14] which combine CNN and LSTM with word embeddings. Experiments showed that combined models can achieve more outstanding performance. Asghar et al. [15] combined two deep learning models, namely BiLSTM and CNN for rumor detection. They experimented different machine learning classifiers as well as deep learning models. The proposed system shows the best result with 86.12% accuracy. Another study on rumor detection and stance classification task was conducted by [16]. They included the user credibility information in their method in addition to the attention mechanism and get good results. In [17] the researchers proposed a CNN model to detect rumors on Twitter. They regulated the hyperparameter settings for their model to achieve the best performance compared to existing machine learning algorithms. They used the publicly available PHEME dataset to train their model. A recent study published on 2021 [18] that

[1] http://www.zubiaga.org/datasets/

investigate the rumor identification problem using the contextual information. They combined CNN and Bi-LSTM and used the Glove embeddings to detect rumors. The experiments are trained on publicly available dataset collected from Kaggle and achieved accuracy of 90.93%. A novel method for rumor detection that learns the implicit features between the main post and the contextual replies. The experiments show that the proposed model is effective for detecting rumors [19].

A comparison study of deep learning approaches used to detect rumor was conducted by [20] show that deep learning architectures are more suitable on rumor detection than the traditional methods. Moreover, most of the feature-based methods are limited and time-consuming. Thus, this paper aimed to investigate the applicability of our model using CNN and LSTM with word embedding for Arabic language. Furthermore, we aim to employ deep learning methods to compare the performance to the traditional methods.

4. Data preparation

4.1 Dataset

We utilized the dataset of Arabic tweets collected by Alzanin et. Al [8]. This dataset contains a total of approximately 270k tweets categorized into 89 rumor stories and 88 non-rumor stories. The tweets collected using Twitter search API and were partially labeled as rumor or non-rumor as the author used a semi-supervised learning model that do not require a huge amount labeled data unlike the supervised learning model. Thus, three Arabic native speaker annotators have labeled the rest of the dataset manually to remove the irrelevant tweets such as conversation, questions, advertisement, etc. Table 1 represents sample of each type of tweet.

4.2 Pre-processing

We perform a series of pre-processing steps on the tweets to prepare the dataset for the model. First, we removed all non-Arabic text, special characters, stop words, punctuations, URLs, and diacritics such as thashdid, damma, kasra, fatha, etc. Second, we applied normalization for letters. For example, replace letter (إآأ) with (ا), letter (ى) with (ي), letter (ة) with (ه), letter (گ) with (ك). Third, we removed all repeating characters for example, the word (سلاااااام) becomes (سلام) to reduce noise.

Table 1 Sample of tweets

Tweet	*Tweet Type*
انتخاب #برهم_صالح رئيسا للعراق Barham Salih elected as a president of Iraq	News
سعدت بانتخاب برهم صالح رئيسا للعراق وسقوط مرشح برزاني I was pleased with the election of Barham Saleh as president of Iraq and the fall of Barzani’s candidate	Conversation
هل انت مع انتخاب برهم صالح رئيسا للعراق Do you support the election of Barham Salih as president of Iraq?	Question
صالح رئيسا للعراق انتصرت ارادة الخير #برهم_ والاصلاح مبارك لنا جميعا ونتمنى ان تكون بداية التغيير الحقيقي نحو عراقنا Barham Salih is president of Iraq, congratulations to all of us, and we hope that this will be the beginning of real change for our Iraq	Wish

4.3 Feature Representation

After pre-processing and cleaning our textual data, we will transform our data to numerical features. The process of representing words as feature vectors in a multidimensional space called word embedding. It can solve the problem of representing data in text classification by finding a suitable numerical representation. It is a powerful way to tokenize by using the dense of word vectors. For deep learning models, we used word embeddings and for machine learning models we used Tf-Idf vectors.

4.3.1 Word Embedding

In neural network, word embedding is one of the best methods of feature representation. Word2Vec [21] is one of the most commonly used word embedding in neural networks. Word2Vec models are trained on large sets of data with lower computational complexity compared to other approaches [22]. AraVec [23] is the Arabic model of Word2Vec that provides multiple dimensions for various text domain such as Twitter, Wikipedia articles, or web pages. Moreover, each model was built using two techniques, Continuous Bag-of-Words (CBOW) and Skip-gram. The CBOW architecture predicts a certain word based on the surroundings words, whereas the Skip-gram predicts the context words based on a certain word. In this work, we use AraVec pre-trained word embeddings with the Twitter dataset and the CBOW architecture.

4.3.2 Tf-Idf Vectorizer

Before training machine learning algorithms, we prepared the dataset using natural language processing techniques such as Term Frequency and Inverse document frequency (TF-IDF) approach. This can represent text data in numeric vectors so the computer can understand [24]. Basically, the TF-IDF technique is one of the most widely used term weighting schemes in information retrieval systems. It is a metric that multiplies the two measures tf and idf. These measures determine the usefulness of terms in describing the document in which they appear. Thus, this technique is used to evaluate the importance of word in a tweet [25].

5. Proposed Models

In this section, we demonstrate the proposed models to detect rumors on Twitter dataset.

5.1 LSTM model Architecture

LSTM is an extension of Recurrent Neural Networks (RNN) [26]. LSTM is efficient at remembering long sequences and modeling long-distance relations. LSTM prevent the problem of gradient vanishing in the RNN and produce better results. Simple architecture of LSTM [27] is shown in figure 2. LSTM is designed to improve performance of RNN as they learn from the past context. The structure of LSTM is formulated as follows:

$$
\begin{aligned}
i_t &= \sigma(U_i h_{t-1} + W_i x_t + V_i c_{t-1} + b_i) \\
f_t &= \sigma(U_f h_{t-1} + W_f x_t + V_f c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t \tanh(U_c h_{t-1} + W_c x_t + b_c) \qquad (1) \\
o_t &= \sigma(U_o h_{t-1} + W_o x_t + V_o c_t + b_o) \\
h_t &= o_t \tanh(c_t)
\end{aligned}
$$

Where (σ) is the logistic sigmoid function, and the i_t, f_t, c_t, o_t are the input, forget, output gates.

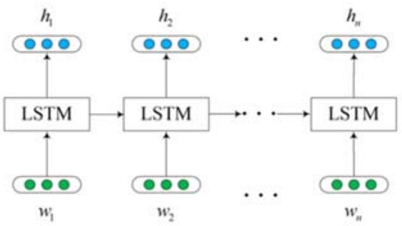

Fig 2 LSTM Model Architecture

5.2 Bi-LSTM Model Architecture

There is an extension of the traditional LSTM model called bidirectional LSTM (BiLSTM). However, the data is processed in both directions with forward LSTM and backward LSTM to improve accuracy by learning from the past and the future [26]. Additional layers are added which called forward hidden layer and backward hidden layer. Simple architecture of Bi-LSTM [27] is shown in figure 3. The structure of LSTM is formulated as follows:

$$
\begin{aligned}
h_t^f &= \tan h\,(W_{xh}^f x_t + W_{hh}^f h_{t-1}^f + b_h^f) \\
h_t^b &= \tan h\,(W_{xh}^b x_t + W_{hh}^b h_{t+1}^b + b_h^b) \qquad (2) \\
y_t &= W_{hy}^f h_t^f + W_{hy}^b h_t^b + b_y
\end{aligned}
$$

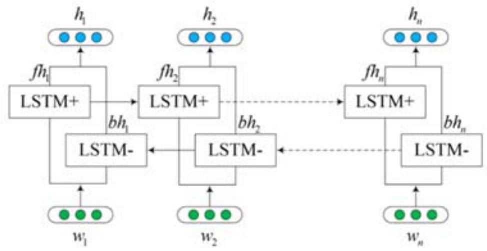

Fig 3 Bi-LSTM Model Architecture

5.3 CNN-LSTM Model

In this section, our hybrid neural network model is proposed for rumor detection. We assume that the use of hybrid method would improve the performance of the model and give much better results on our classification task. Using a combined models allows the network to extract local and deep features from the CNN, so the LSTM layers take them as input [28]. Recently, CNN models have achieved good results when used with text data for classification tasks [29], [30]. CNN layers are helpful in filtering out the noise of the input data. Our proposed model consists mainly of convolutional layer and a recurrent layer. The fundamental architecture of our CNN-LSTM model is shown in Figure 4. The details of each layer in the model are described as follows:

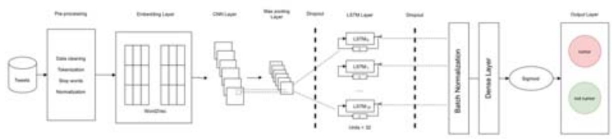

Fig 4 CNN-LSTM Model Architecture

5.3.1 Embedding Layer

The first layer in our neural network model is the embedding layer that accepts the input as a vector of 100 words. A fixed dimensional vector is mapped onto each word. The tweets are encoded using a pre-trained word embedding on Twitter dataset with a CBOW model by 300-dimensional vectors.

5.3.2 Convolutional Layer

The role of convolutional layer is to provide semantic features to the text inputs. We used a one-dimensional convolutional layer associated with a filter window k = 5 with the non-linear ReLU activation function. Followed by a Max-pooling layer to minimizes the number of features and operations for computing the output for the following layers. Thus, it can reduce the computational efforts during feature learning. This function returns the maximum set of values in a rectangular area from the output of the previous layer [31].

5.3.3 Dropout Layer

The main purpose of dropout layer is to drop random units to reduce overfitting in neural networks. A dropout is a technique used for reduce overfitting in the neural networks that have a set of parameters [32]. The likelihood of retaining a hidden unit in the network can be determined using hyperparameter tuning in the dropout layer [33]. In our model, a dropout layer with a value of 0.3 is added after each model to improve our neural network model. As a result, this will increase the accuracy, and loss will gradually decrease because of the dropout value.

5.3.4 LSTM Layer

An LSTM layer has been added to handle the nature of sequential data by considering the current information with the past information. LSTM keep long-term memory to update the previous hidden layers. This layer contains one hidden layer with 10 units.

5.3.4 Fully Connected Layer

We used a batch normalization technique to improve and accelerate the training of a neural network model by maintain the dissemination of previous layer [34]. Then we have a dense layer to connect every input to every output by some weight. In this layer, we passed a Sigmoid function to the output layer to give the final classification probability. We have taken Sigmoid [35] as the activation function because it is capable of to solve binary classification problems.

5.4 Baseline Models

For baseline model, we used various supervised machine learning algorithms to evaluate the performance of our hybrid model to detect rumors. That's include K-Nearest Neighbors (KNN), Gradient Boosting (GB), and Extreme Gradient Boosting (XGBoost). KNN is classified as memory-based approach because is memorizes the training dataset instead of learning the discriminative functions [36]. GB is a powerful machine learning model with a high customizable capability that aims to improve the performance by reducing errors as it is a gradient-based classifiers that can be used for regression or classification problems [37]. Where the XGBoost is more efficient implementation of gradient boosting model [38].

6. Experiments and Results

To conduct the experiments, we split the dataset into training set and testing set (80:20) for both machine learning and deep learning classifiers. The following hyper-parameters were optimized after experimenting with several trials to choose the best parameters to yield the best performance results. In order to measure the performance of a binary classification model, we set the loss function value as binary cross-entropy. As for the adaptive learning optimizer, we have taken Adam algorithm for training our algorithm.

Table 2 Hyper-parameter settings for all classifiers

Model	***Hyper-parameter***
KNN	n-neighbors:5, weights= uniform, leaf_size:30, p:2, metric:minkowski
GB	Criterion: friedman_mse, learning rate: 0.1, n_estimators: 100, loss: deviance
XGBoost	Objective: binary:logistic, Max_depth: 3, learning rate: 0.1, gamma:0, subsample: 1, n_estimators:100
LSTM	Max features(word2vec): 99503700, dropout: 0.2 , epochs:10 , batch size: 32 , embedding size: 300, units:10
Bi-LSTM	Max features(word2vec): 99503700, dropout: 0.3 , epochs:10 , batch size: 32 , embedding size: 300 , activation: sigmoid
CNN-LSTM	Max features(word2vec): 99503700, dropout: 0.3 , epochs:10 , batch size: 32 , embedding size: 300, kernel size: 5, filter: 10, max pooling: 4, activation function: relu, lstm units: 10, activation: sigmoid

6.1 Evaluation Metrics

To evaluate our models, we used the following evaluation metrics: Accuracy, Recall, Precision, and F1-score. See formula 3, 4, 5 and 6 where TP, TN, FP, and FN indicate true positive, true negative, false positive, and false negative.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall} \quad (6)$$

The achieved results are shown in table 3. In order to evaluate the performance of our proposed model for rumor detection, we compare our model with a basic machine learning classifier (KNN) in addition to several deep learning models. As it can be seen, the deep learning models outperforms the traditional machine learning algorithms. Combining CNN with LSTM show significant results that outperforms other models. To summarize, our results show that our approach perform better when adding convolutional layers. The model proposed in this paper has improved performance compared to the existing models and obtained the best performance of 95.91%. The lowest result out of all classifiers is KNN with accuracy of 72.97%.

Table 3 Summary of results

Model	*Accuracy*	*Precision*	*Recall*	*F-score*
KNN	72.97	79.24	77.81	72.92
GB	88.22	90.81	85.97	87.22
XGBoost	87.49	90.39	84.30	85.95
LSTM	92.32	87.27	92.73	89.91
Bi-LSTM	92.77	85.98	95.13	90.32
CNN-LSTM	**95.91**	**97.35**	92.95	**94.91**

6.2 Error Analysis

After the evaluation phase, we used the confusion matrix to get a better understanding of the results of predictions. A confusion matrix is a table layout that shows visualization of the performance of any supervised learning algorithm [39]. Figure 6 shows the confusion matrix of our proposed model CNN-LSTM. We can observe from figure 4 that our model predicted 2209 out of 3766 tweets as true positive which mean a total of 2209 were predicted as non-rumor correctly. As for the true negative, the model predicted 1399 instances out of 3766 as rumor correctly. While the model predicted 80 tweets out of 3766 as non-rumor, but they were not. Moreover, the model wrongly predicted 78 tweets as rumor, but it is actually non-rumor.

6.3 Training Loss and Performance

Based on the hyper-parameter tuning settings, the optimum parameter for batch size was 32 with the epochs 10. Adding more data might increase the accuracy of the model.

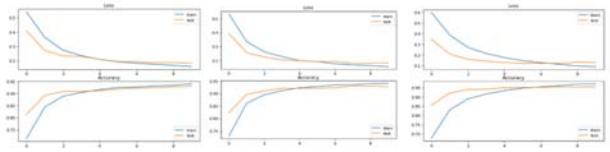

Fig 5 Learning Curve for neural network models

Figure 5 illustrates the learning curve for training loss and accuracy for the proposed neural network models LSTM, BiLSTM, and CNN-LSTM respectively. As we can see, a constant increase in of accuracy in all models for both training sets and testing sets. As for the loss curves, we can observe that the training and testing loss decrease together to a stable point. Furthermore, the generalization gap between the training curve and the validation curve is small which indicates a good fit for our applied models.

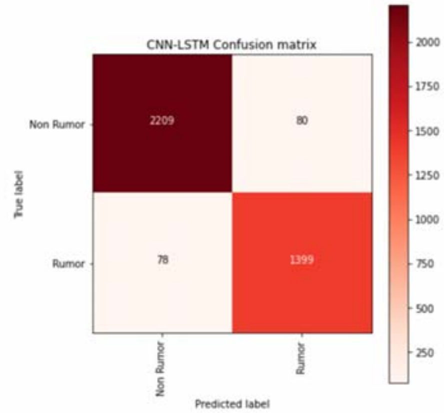

Fig 6 Confusion Matrix for CNN-LSTM model

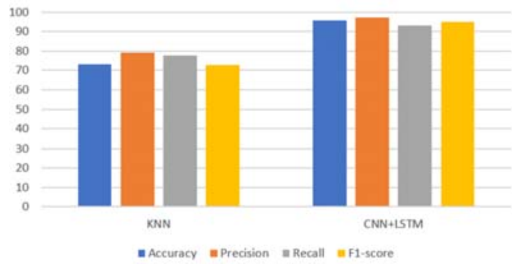

Fig 7 Comparison between our approach and baseline approach based on KNN

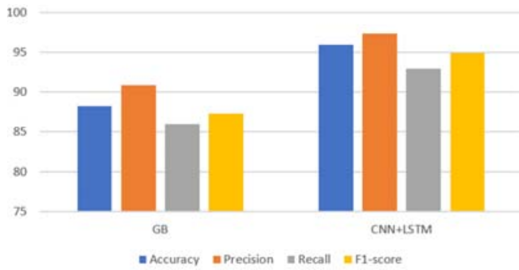

Fig 8 Comparison between our approach and baseline approach based on GB

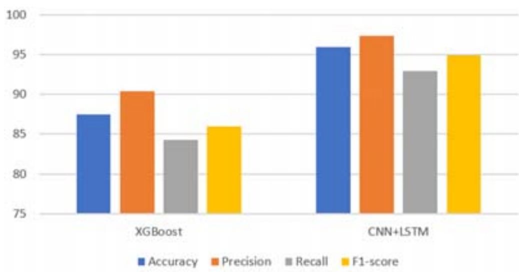

Fig 9 Comparison between our approach and baseline approach based on XGBoost

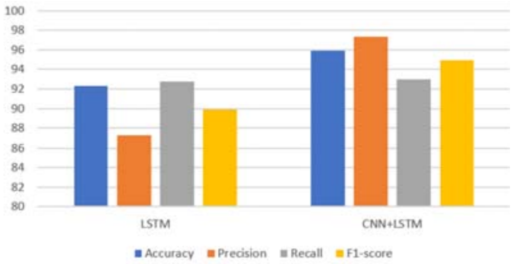

Fig 10 Comparison between our approach and baseline approach based on LSTM

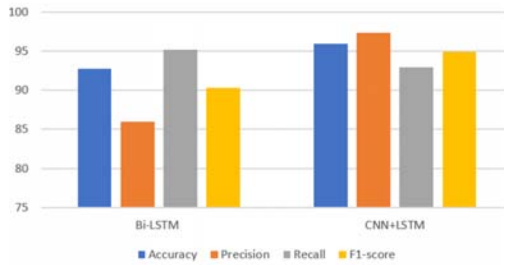

Fig 11 Comparison between our approach and baseline approach based on Bi-LSTM

In figures 7-11, the performance of our proposed model is compared with different baseline approaches. We can observe that the performance of our hybrid model CNN-LSTM to detect rumor on Twitter dataset is outperform the baseline approaches. As shown in the previous figures, we can demonstrate that the accuracy and f-score have increased 3.14% and 4.59% respectively compared to the higher baseline results. Therefore, this implies that the performance of hybrid neural network model using AraVec pre-trained word embedding is effective for the rumor detection task.

7. Conclusion and Future Work

In this paper, we address the problem of detecting rumors on Arabic social media using deep learning models. To this end, the discussed results presented in the previous section show that neural network models perform reasonably well in the rumor detection task. Deep learning models outperform machine learning methods in most cases. Furthermore, it is important to highlight the difficulty of configuring all the parameters of the neural networks, and how slow the training process can be. The performance of different types of neural networks have been evaluated on this paper. Our proposed approach reaches an accuracy of 95.9% for classifying tweets into rumor or non-rumor.
In the future work, we aim to build or extend the used dataset in order to examine our model on a comparatively large number of tweets. Another perspective at future work is using the AraBERT model and learn how it work would be also interesting. Moreover, we want to investigate other deep learning techniques for classifying rumors.

References

[1] E. Shearer and A. Mitchell, "Platforms in 2020 News Use Across Social Media," *Pew Research Center*, 2021. .

[2] S. F. Sabbeh and S. Y. Baatwah, "Arabic news credibility on twitter: An enhanced model using hybrid features," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 8, 2018.

[3] R. H. Knapp, "A psychology of rumor," *Public Opin. Q.*, vol. 8, no. 1, 1944, doi: 10.1086/265665.

[4] P. Meel and D. K. Vishwakarma, "Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities," *Expert Systems with Applications*, vol. 153. 2020, doi: 10.1016/j.eswa.2019.112986.

[5] E. Seo, P. Mohapatra, and T. Abdelzaher, "Identifying rumors and their sources in social networks," in *Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR III*, 2012, vol. 8389, doi: 10.1117/12.919823.

[6] "scholar (5)." .

[7] A. Y. M. Floos, "Arabic Rumours Identification By Measuring The Credibility Of Arabic Tweet Content," *Int. J. Knowl. Soc. Res.*, vol. 7, no. 2, 2016, doi: 10.4018/ijksr.2016040105.

[8] S. M. Alzanin and A. M. Azmi, "Rumor detection in Arabic tweets using semi-supervised and unsupervised expectation–maximization," *Knowledge-Based Syst.*, vol. 185, 2019, doi: 10.1016/j.knosys.2019.104945.

[9] H. K. Thakur, A. Gupta, A. Bhardwaj, and D. Verma, "Rumor Detection on Twitter Using a Supervised Machine Learning Framework," *Int. J. Inf. Retr. Res.*, vol. 8, no. 3, 2018, doi: 10.4018/ijirr.2018070101.

[10] R. K. Kaliyar, A. Goswami, and P. Narang, "A hybrid model for effective fake news detection with a novel COVID-19 dataset," in *ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, 2021, vol. 2, doi: 10.5220/0010316010661072.

[11] X. Chen, L. Ke, Z. Lu, H. Su, and H. Wang, "A novel hybrid model for cantonese rumor detection on twitter," *Appl. Sci.*, vol. 10, no. 20, 2020, doi: 10.3390/app10207093.

[12] L. Zhang, J. Li, B. Zhou, and Y. Jia, "Rumor Detection Based on SAGNN: Simplified Aggregation Graph Neural Networks," *Mach. Learn. Knowl. Extr.*, vol. 3, no. 1, 2021, doi: 10.3390/make3010005.

[13] T. Bian *et al.*, "Rumor detection on social media with bi-directional graph convolutional networks," 2020, doi: 10.1609/aaai.v34i01.5393.

[14] M. Shi, K. Wang, and C. Li, "A C-LSTM with word embedding model for news text classification," 2019, doi: 10.1109/ICIS46139.2019.8940289.

[15] M. Z. Asghar, A. Habib, A. Habib, A. Khan, R. Ali, and A. Khattak, "Exploring deep neural networks for rumor detection," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 4, 2021, doi: 10.1007/s12652-019-01527-4.

[16] Q. Li, Q. Zhang, and L. Si, "Rumor detection by exploiting user credibility information, attention and multi-task learning," 2020, doi: 10.18653/v1/p19-1113.

[17] A. Alsaeedi and M. Al-Sarem, "Detecting Rumors on Social Media Based on a CNN Deep Learning Technique," *Arab. J. Sci. Eng.*, vol. 45, no. 12, 2020, doi:

10.1007/s13369-020-04839-2.

[18] N. Rani, P. Das, and A. K. Bhardwaj, “A hybrid deep learning model based on CNN-BiLSTM for rumor detection,” 2021, doi: 10.1109/icces51350.2021.9489214.

[19] A. P. Ben Veyseh, M. T. Thai, T. H. Nguyen, and D. Dou, “Rumor detection in social networks via deep contextual modeling,” 2019, doi: 10.1145/3341161.3342896.

[20] 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. 2019, doi: 10.1109/ACCESS.2019.2947855.

[21] K. W. CHURCH, “Word2Vec,” *Nat. Lang. Eng.*, vol. 23, no. 1, 2017, doi: 10.1017/s1351324916000334.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013.

[23] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, “AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP,” in *Procedia Computer Science*, 2017, vol. 117, doi: 10.1016/j.procs.2017.10.117.

[24] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” 2015, doi: 10.3115/v1/p14-5010.

[25] A. Aizawa, “An information-theoretic perspective of tf-idf measures,” *Inf. Process. Manag.*, vol. 39, no. 1, 2003, doi: 10.1016/S0306-4573(02)00021-3.

[26] S. Hochreiter and J. Schmidhuber, “Long Short Term Memory. Neural Computation,” *Neural Comput.*, vol. 9, no. 8, 1997.

[27] J. Xie, B. Chen, X. Gu, F. Liang, and X. Xu, “Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification,” *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2957510.

[28] X. Wang, W. Jiang, and Z. Luo, “Combination of convolutional and recurrent neural network for sentiment analysis of short texts,” 2016.

[29] “Scholar (7).” .

[30] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, “Detecting Offensive Language on Arabic Social Media Using Deep Learning,” 2019, doi: 10.1109/SNAMS.2019.8931839.

[31] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” in *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, 2014, vol. 1, doi: 10.3115/v1/p14-1062.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, 2014.

[33] A. S. Walter *et al.*, “Scholar (8),” *Convergence in the information industries. Telecommunications, broadcasting and data processing 1981-1996*, vol. 3, no. 1. pp. 125–150, 2004, [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0160738315000444%250Ahttp://eprints.lancs.ac.uk/48376/%255Cnhttp://dx.doi.org/10.1002/zamm.19630430112%250Ahttp://www.sciencedirect.com/science/article/pii/S0160738315000444%250Ahttp://eprints.lancs.ac.uk/48376/%255.

[34] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” in *Advances in Neural Information Processing Systems*, 2018, vol. 2018-December.

[35] J. Han and C. Moraga, “The influence of the sigmoid function parameters on the speed of backpropagation learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1995, vol. 930, doi: 10.1007/3-540-59497-3_175.

[36] Y. Ren, “Python Machine Learning : Machine Learning and Deep Learning With Python ,” *Int. J. Knowledge-Based Organ.*, vol. 11, no. 1, 2021.

[37] “scholar (9).” .

[38] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, 2016.

[39] V. M. Patro and M. Ranjan Patra, “Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy,” *Trans. Mach. Learn. Artif. Intell.*, vol. 2, no. 4, 2014, doi: 10.14738/tmlai.24.328.