Journal of Theoretical and Applied Information Technology
31st July 2022. Vol.100. No 14
© 2022 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AFND: ARABIC FAKE NEWS DETECTION WITH AN ENSEMBLE DEEP CNN-LSTM MODEL

SHAYMAA E. SOROUR[1], HANAN E. ABDELKADER[2]

[1] Faculty of Specific Education, Kafrelsheikh University, Kafrelsheikh, Egypt

[2] Faculty of Specific Education, Mansoura University, Mansoura, Egypt

E-mail: [1] shaymaa.srour@spe.kfs.edu.eg, [2] haanan277@mans.edu.eg

ABSTRACT

The rapid expansion of misinformation in daily life has disrupted different news sources, such as social media, online news, radio and television stations, and newspapers, making it difficult to choose reliable news outlets. The potential to spread fake news (FN) to many organizations and platforms jeopardizes news credibility and causes users to abandon them. However, detecting FN entails predicting the probability that a particular news article is deceptive or not. However, most contemporary methods do not consider Arabic news and how Arabic FN (AFN) has been detected in the past decade. Therefore, research on AFN detection is beginning to receive more attention. This paper presents an Arabic FN detection (AFND) system based on hybrid deep learning (DL) model. This model includes both conventional neural network and long short-term memory (CNN-LSTM) modalities. The input dataset was prepared via discretization and normalization. Then, word vectors were included with the corrected words at a given word length as pretrained vectors on Arabic news. Due to outstanding performance, the JSO optimization algorithm was combined with the framework to automatically define the best structure for the proposed CNN-LSTM. A comparison was made between the proposed CNN-LSTM and other recent models to prove the performance of the proposed CNN-LSTM. The results indicate that the proposed CNN-LSTM offers the best performance, with an accuracy of 81.6%. The experimental results provided comprehensive improvements in the subject matter of AFND and demonstrated the potential of the proposed methodology.

Keywords: *Arabic Fake News Detection (AFND), Deep Learning (DL), Conventional Neural Network (CNN), Ensemble Learning, Optimization, Long Short-Term Memory (LSTM).*

1. INTRODUCTION

The Internet has become an integral part of our daily lives as its users increase every day. As a result, social media (SM) has proliferated in scale, resulting in a significant increase in using people's read blogs, news articles, and news content from different sources every day. Due to the different sources, it is easy to be distracted and misled by any misrepresentation of an event or event that does not exist but has been portrayed in a verified style [1, 2]. Currently, fake news (FN) and disinformation have been used as a weapon to achieve unethical goals. FN is defined as "a claim or information proven to be false"[3, 4]. Nowadays, misinformation is a major problem as it quickly spreads and reaches numerous people.

However, manual methods for detecting FN are costly and time-consuming. Therefore, there is a need for cheap and fast methods that can detect FN automatically, limiting the spread of dubious content and disabusing the public in the case of false news [5]. The prevalence of such FN has a profound negative impact on both the target individuals and society. In addition, it creates an impression on readers, impairing public perception and responses to authentic news and threatening the balance of the news ecosystem. One startling example is the 2016 US presidential election, where FN circulated more than real news on Facebook and Twitter [6].

Furthermore, as the COVID-19 pandemic continues to spread, misleading or false information about COVID-19 could become a serious problem, with negative consequences in a variety of fields, such as economy, labor market, industry, entertainment, journalism, and education[7]. Technological advancements have made accessing real and fake information easier, thus posing a real challenge. SM is used instead of traditional media as it has a greater motivational effect, although fake and real news spreads very quickly. Thus, it is difficult for SM platforms, such as Twitter, WhatsApp, YouTube, and Facebook, to distinguish between FN and real news within the massive number of users' posts. Hence, there is a risk involved in publishing and disseminating news on SM platforms[8].

The study on FN is attracting an increasing amount of active research from academia and industry. Due to the advent of technology, specialized knowledge is applied to extract hand-crafted features from news text content, which are then used to train an FN classifier using traditional machine learning algorithms. These methods, which rely on hand-crafted features, are very simple yet limited in scope and flexibility[9, 10]. Extensive research indicates that the ability to design synthetic features is critical for various types of natural language processing tasks. By automatically extracting features from news text content, deep learning (DL) technologies have provided new insights into fake news detection (FND) and achieved cutting-edge performance[11]. Extensive research indicates that the ability to design synthetic features is critical for various types of natural language processing tasks. By automatically extracting features from news text content, deep learning (DL) technologies have provided new insights into fake news detection (FND) and achieved cutting-edge performance.

This study determines whether current developments in DL models and extensive linguistic models of Arabic can be effectively applied to detect Arabic FN detection (AFND) task.

FN identification can be considered a classification problem; this means that we want to know if a particular tweet is fake or real. This research will investigate the use of DL forms for AFND assignment through the Arab news group (ANS) application[12].

The goal of this study is to evolve an AFND system with the highest values in terms of evaluation criteria such as F-measure and accuracy. To achieve this aim, the proposed model has been utilized for detecting AFN. The system is splitted into four phases. During the preprocessing phase, the word vectors are generated from the corpus as pretrained vectors to analyze Arabic sentiment, remove all errors, and use the under-sampling method. The jSO optimizer automatically generates optimal hyperparameters for the proposed model in the second stage. Then, the classification tasks are conducted using a hybrid DL model comprising both conventional neural network and long short-term memory (CNN-LSTM) modalities. In the final phase, the proposed model is evaluated using a variety of evaluation metrics to check its performance. The following are the main contributions of this work:

- Preprocessing the AFNs data.
- The SO optimizer is used to automatically generate optimal hyperparameters for the proposed CNN-LSTM.
- The proposed hybrid model comprises both CNN and LSTM DL modalities for the conducted classification tasks.
- The proposed CNN-LSTM is evaluated using various evaluation metrics to check its performance.
- The performance of the proposed model is compared with those of the latest technologies and evaluated.

The rest of this paper is structured as follows. The literature review is summarized in Section 2. Section 3 discusses the background. Section 4 presents the proposed model. Section 5 contains a discussion and analysis of the experimental results. Finally, Section 6 wraps up the paper by discussing the implications of the findings and suggesting areas for future work.

Nomenclature

AFNs	Arabic Fake News
AFND	Arabic Fake News Detection
ANS	Arabic News Stance corpus
CNN	Conventional Neural Network
CBOW	continuous vector representations
DL	Deep Learning
DE	Differential Evolution Algorithm
FND	Fake News Detection
FNs	Fake News
ML	Machine Learning
LSTM	Long Short-Term Memory
SM	Social Media
n	the total number of records in the dataset

D number of decision variables
$x_{i,j}^{min}$ and $x_{i,j}^{max}$ the lower and upper bounds at each j_{th} dimension
n_{fes} current number of function evaluations
max_{nfes} maximum number of function evaluations
σ, g, and h the gate, input, and output activation functions
$f_i(w)$ loss of I record of the dataset
l_t vector of activation of each cell
v_t vector of activation of each memory block at time t
x_t input data at time t
i_t, f_t, o_t input, forget and output gates at time t
z_t output data at time t
q weight coefficients
P Precision
R Recall
F_1 F-measure
acc Accuracy

2. LITERATURE REVIEW

This section provides an overview of FND based on SM rumors, focusing on rumor mining and analysis of linguistic features of the claim to determine their validity without looking at external factual information. The detection of FN and classification of news data based on their degree of falsity were accomplished mainly by using word embedding, GloVe embedding, for example, can be used to extract the semantic meanings of words, after which the metadata information is added to the model either by arranging its static representations of the sentence or by using attention mechanisms[13-15].

Furthermore, textual representations are primarily expressed and modeled using the tensor factor[16]. and deep neural methods[17], both of which are effective in identifying FN. To extract the different features of FN, visual properties are captured from visual elements, such as recordings and images[18]. In most contextual studies, researchers investigated the issue of FN using a different dataset. For example, Akhtar et al.[4] introduced a new approach to assist determine whether a message about COVID-19 is fake or real. The dataset included 10,700 real and fake COVID-19 SM posts and news articles in English. They combined bidirectional LSTM, support vector machine, logistic regression, and Naive Bayes and used a combination of logistic regression and Naive Bayes to create an ensemble.

Veyseh et al[19] examined a semantic graph approach for rumor detection by modeling semantic relationships between key posts and responses based on their content. This model learns the implicit relationships between the main tweet and its responses. They used Twitter datasets to compare the results to the most recent rumour detection methods described in [20]. The proposed model was evaluated in comparison to feature- and DL-based models. The results showed that the DL models outperformed the feature-based models in detecting rumours. They also proved that the semantic graph approach achieved best-in-class accuracy in the two datasets by integrating implicit semantic relationships between all tweets in the thread. Ghanem et al.[21] presented the fake-flow model to determine whether a news is fake. The model was based on the idea that the articles for FN often capture the reader's attention with emotional appeals that elicit their feelings. They used neural architecture to model the effective information flow in FN articles. Fake-flow learns the flow by combining the topic with effective information extracted from the text. CNN, bidirectional gateway recurrent modules, hierarchical attention networks, LSTM, bidirectional deep transformer pretraining (BERT), and Longformer were compared with the proposed fake-flow. The fake-flow achieved 96%, 93%, 97%, and 96% accuracy, precision, recall, and macro F_1 scores, respectively. The Longformer model outperformed with a macro F1 score of 97%.

Shim et al. [22] suggested a new embedded method named link2vec, which is an extension of Word2vec. They applied their link2vec model to two real-world FND datasets in two different languages (Korean and English). In both datasets, the experimental results indicated that the link2vec method statistically outperformed the others. Jain et al. [23] proposed an efficient DL algorithm for detecting the degree of counterfeiting in a news report. The efficiency and effectiveness of the algorithms have been confirmed in multiple real-world datasets. Their method achieved a classification accuracy of 46.36% in the LIAR dataset, outperforming the state-of-the-art technology by 1.4%.

Based on our review of relevant work in the field of AFND, Jude Khouja proposed AFND based on the ANS dataset. They implemented pretraining (BERT) and LSTM models and obtained mean overall F-measures of 64% and 40%, respectively. Al-Yahya et al. [3] provided a comprehensive comparison of neural network and transformer-based language models utilized in AFND. They used the ANS dataset and obtained mean overall F-measures for AraBERT v02, QARiB, and AraGPT2 of 3%, 6%, and 20%, respectively. They also applied the proposed models to the COVID-19 FAKES dataset and obtained mean overall F-measures for AraBERT v02, QARiB, and AraGPT2 of 67%, 56%, and 61%, respectively. Research and investigation on AFND are clearly not available, which necessitates further research and investigation. Consequently, this study sheds light on new model-based approaches to the AFND task.

3. BACKGROUND

The following section provides the main concepts of the CNNs, LSTM and jSO optimization algorithms.

3.1 Convolutional Neural Networks (CNN)

CNN is a deep learning algorithm that learns spatial hierarchies of features ranging from low-level to high-level patterns automatically and adaptively. Convolutional, pooling, and fully connected layers are the three types of building blocks that comprise a CNN. The first two layers, convolution and pooling, extract the features, while the third layer, a fully connected layer, maps the extracted features into a final output, such as classification. A convolution layer [24] is required for CNN, which is made up of a stack of arithmetic operations such as convolution, which is a type of linear operation. Convolutional processing seeks to extract features from data by dividing a row of data into small subsets and performing the same operations on each subset.

This process relies heavily on the kernel and strides. This procedure is heavily reliant on the kernel and strides. Each element of the mini-set will be multiplied by the kernel, which is a collection of constant values. The kernel size k determines the size of the mini-set. The elements of the corresponding sin wave arraay, and the output Y is expressed in Eq. 1:

$$Y = \sum_{i=0}^{2} k_i x_i \qquad (1)$$

This is referred to as valid convolution. The shape of the output is calculated using Eq. 2:

$$\left(\left[\frac{L-C}{T}+1\right],\left[\frac{R-C}{T}+1\right]\right) \qquad (2)$$

where the data of shape (L,R), the kernel of shape (C,G), and the stride is equal to T. In deep neural networks, the Leaky ReLU activation function is widely used. It assists to solve the vanishing gradient problem and not activate all neurons at the same time [25]. In the final dense (fully connected) layer before the output layer, the Leaky ReLU activation function is also used. The Sigmoid activation function from Eq. 3 is used for binary classification in the output layer.

$$f(z) = \frac{1}{1+e^{-z}} \qquad (3)$$

The loss function is computed using the model output and data labels; the goal of the training phase is to minimize the loss value. Because the model's binary output is either zero or one, the loss is known as the binary cross-entropy, as shown in Eq. 4:

$$l = y\ \log(y') + (1-y)\log(1-y') \qquad (4)$$

$$H = \sum_{i=0}^{n} l_i \qquad (5)$$

where H is the overall loss, l is the loss for one entry, y is the entry label, y' is the predicted label for this entry, and n is the number of entries in the dataset.

3.2 Long Short-Term Memory (LSTM)

LSTM [26, 27] is a type of RNN that has the power and efficiency to tackle both short and long term dependency problems. The memory cell, which exchanges the hidden layers of traditional neurons, serves as the basis for the LSTM network. Because it has three exchanges (input, forget and output gates), it can add or delete information from the state of the cell. The technique for updating the state of the cell and calculating the output from the LSTM model is defined by Equations 6 - 11[27, 28].

$$r_t = \sigma_g(q_{ix}x_t + q_{im}m_{t-1} + q_{il}l_{t-1} + b_i) \qquad (6)$$

$$y_t = \sigma_g(q_{yx}x_t + q_{ym}m_{t-1} + q_{yl}l_{t-1} + b_y) \qquad (7)$$

$$l_t = f \square\ c_{t-1} + i_t \square\ g(q_{cx}x_t + q_{lm}m_{t-1} + b_l) \qquad (8)$$

$$m_t = \sigma_g(q_{ox}x_t + q_{om}m_{t-1} + q_{ol}l_{t-1} + b_o) \qquad (9)$$

$$v_t = O_t \square\ \sigma_h(l_t) \qquad (10)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$f_t = q_{fm} m_t + b_f \quad (11)$$

Where o_t and f_t denote the input and output data, respectively, at time t . r_t , y_t , and m_t are the input, forget, and output gates at time t , l_t is the activation vector of each cell, and v_t is the activation vector of each memory block. Gate, input, and output activation functions are represented by σ , g , and c . The weight coefficients are denoted by q .

3.3 JSO Optimization Algorithm

The jSO algorithm [29] is based on the iL-SHADE, an improved L-SHADE variant in which L-SHADE uses a differential evolution algorithm (DE) [29] to use a population linear reduction approach. DE consists of four main operations: initialization, mutation, crossover, and selection, as well as three parameters: population size (NP), crossover rate (CR), and scaling factor (F). Finding the best configuration of the three parameters by hand is typically a time-consuming task. As a result, one recent self-adaptation technique was used in this study to address the problem of selecting the best hyperparameters values[29]. The major steps in the jSO algorithm are described in Algorithm 1. This algorithm also employs a linear population size mechanism to update the individuals in the entire population for every generation [30].

Algorithm 1 jSO algorithm

```
Define $g \leftarrow 0$ , $A \leftarrow \{\}$ ;
Generate an initial random population $(P_0)$ of size $NP$ ;
Estimate $f(P_0)$ , and update number of fitness evaluations $fes \leftarrow fes + NP$ ;
while $fes \le max_{fes}$ do;
    $g \leftarrow g + 1$ ;
    for $i = 1 : NP$ do;
        Generate solution $(\vec{v}_i^{\,g})$ using [30];
        Employ crossover to generate off spring solution $(\vec{u}_i^{\,g})$ using [30];
        Estimate new population $f(\vec{u}_i^{\,g})$
    end for
    for $i = 1 : NP$ do
        if $f(\vec{u}_i^{\,g}) \le f(\vec{x}_i^{\,g})$ then
            $\vec{x}_i^{\,g+1} \leftarrow \vec{u}_i^{\,g}$ ;
        else
            $\vec{x}_i^{\,g+1} \leftarrow \vec{x}_i^{\,g}$ ;
        end if
        if $f(\vec{u}_i^{\,g}) < f(\vec{x}_i^{\,g})$ then
            $x_i^g \rightarrow A$ ;
            Update $F$ , $Cr$ , and $F_w$ using [30];
        end if
        Update $A$ if needed;
    Employ linear population size reduction technique using [30];
    end for
end while
```

Where NP is the population size, D the dimensionality of the problem, and $x_{i,j}^{min}$ and $x_{i,j}^{max}$ the lower and upper boundaries at each j^{th} dimension, respectively. The maximum number of function evaluations and current ones are denoted by max_{fes} , and n_{fes} .

4. THE PROPOSED MODEL

This section contains a detailed description of the proposed framework. The main goal is to build an AFND model that includes both CNN and LSTM methods to achieve the highest accuracy results during the experiment. As a result, the CNN-LSTM model was used to identify the most relevant features from a given data set and enable the system to repeatedly assess the relevance of the selected features in terms of accurate prediction of fake or real news. Figure 1 depicts the AFND framework which contains four steps: (1) input data, (2) data preprocessing, (3) jSO optimizer, and (4) utilizing CNN- LSTM sequential model.

4.1 Input Data

The dataset applied in the current research is called ANS [12]. The ANS is a collection of Arabic news headlines.

4.2 Data Preprocessing

Data pre-processing is a critical step in preparing the data for the proposed model. Data preprocessing involves a number of critical steps, such as removing stop words, sign (e.g., ",", "’",„ ".","!","?"), punctuation, discrimination, normalization, and then generation of word vectors (embeddings) from the corpus of sentiment analysis as pre-trained vectors on the Arabic news model 2[1]; It is a collection of Arabic

[1] https://github.com/iamaziz/ar-embeddings

materials containing a mixture of formal and spoken Arabic (dialect)[31].

To learn about word embeddings, Arabic-news models utilized word2vec model. The skip-gram and Continuous Bag-Of-Words architectures are described in Word2vec as two architectures for computing continuous vector representations (CBOW). The former predicts context words based on a given source word, while CBOW predicts a word based on its context window. Since it is simpler, more computationally efficient, and suitable for larger datasets, CBOW has been used for embeddings learning. This brings us to the three stages of data pre-processing, which are as follows:

- Scan all words and sentences by correcting spelling of words and spaces between words. Figure 3 shows examples of the out of the wrong_word and the correct_word. Remove all errors and save data.
- Sort data by length; the longest at the top and the smallest at the bottom.
- Apply under-sampling technique to balance uneven datasets by retaining all data in the minority class while decreasing the size of the majority class. It is able to ex- tract more accurate information from initially unbalanced datasets[32].

4.3 JSO Optimizer

The jSO optimization algorithm is used to adjust the hy- perparameters of both CNN and LSTM in order to select the best structure during the running stage and to accurately predict AFNs. jSO starts with an initialization step in which a number of CNN and LSTM structures (hyperparameters) are randomly generated and their training accuracy (fitness value estimated by the validation loss) is computed. Then, the population as a whole (CNN-LSTM structures) is evolved to generate new solutions (new CNN-LSTM structures). At the end of each generation, a greedy selection method is used to determine which CNN-LSTM structure will enter the new population. A a Population Linear Reduction Technique (PLRT) is also performed at the end of each generation to ensure a balance of exploitation and exploration. This process will continue until the stopping condition is met. The proposed model is compared to other algorithms that do not use jSO and is shown to be superior.

4.4 CNN-LSTM Sequential Model

The CNN-LSTM sequential DL model is proposed in this paper for AFN detection. LSTM is well-suited for classification- tion and processing. The relative gap length sensitivity gives LSTM an advantage over other Recurrent Neural Networks (RNNs), hidden Markov models, and sequential learning methods. Figure 3 presents the main activities of the proposed sequential CNN-LSTM model. The proposed model consists of one convolutional layer used for features extracting from the text embeddings followed by one max pooling layer for dimension reduction. The convolutional layers have a number of filters of 64 and kernel size of 3, followed by LSTM layer to process the sequence of features as a time series, with 64 units, dropout rate of 0.

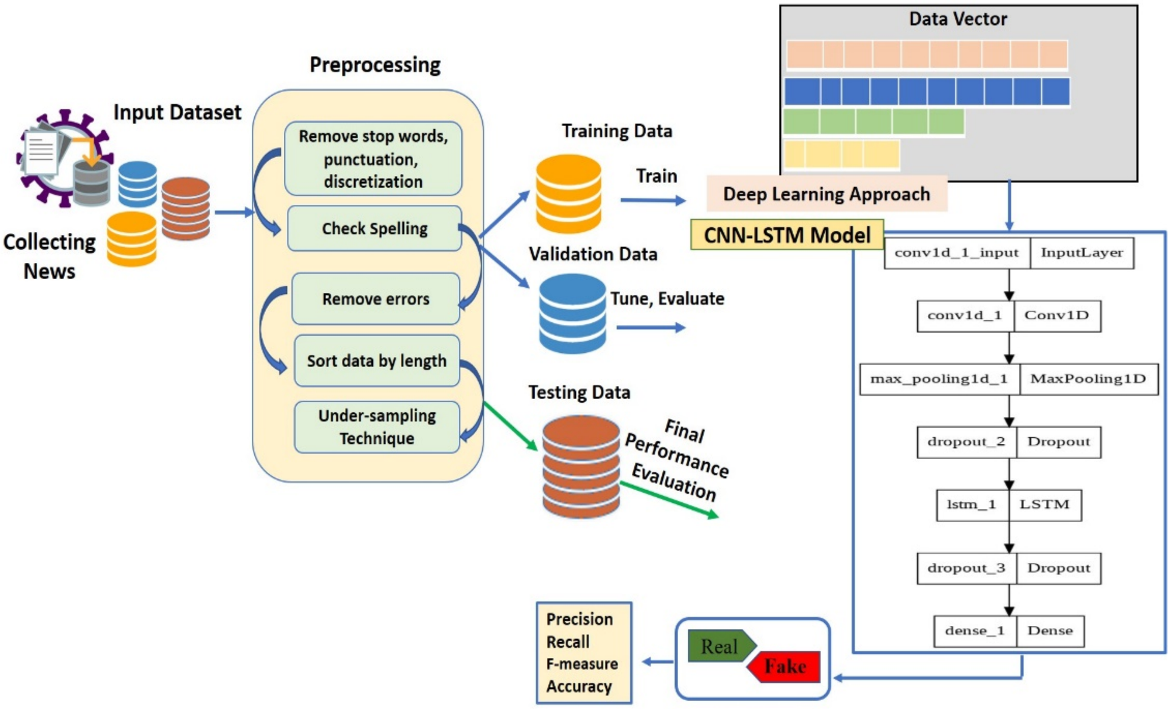


Figure 1: The proposed Arabic Fake News detection framework.

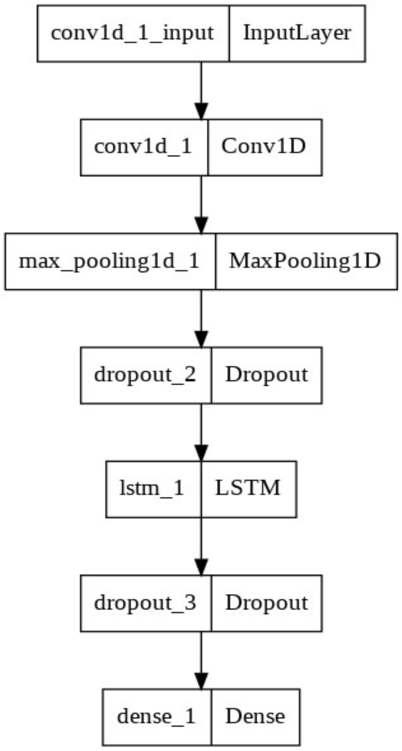


Figure 2: The proposed CNN-LSTM Model.

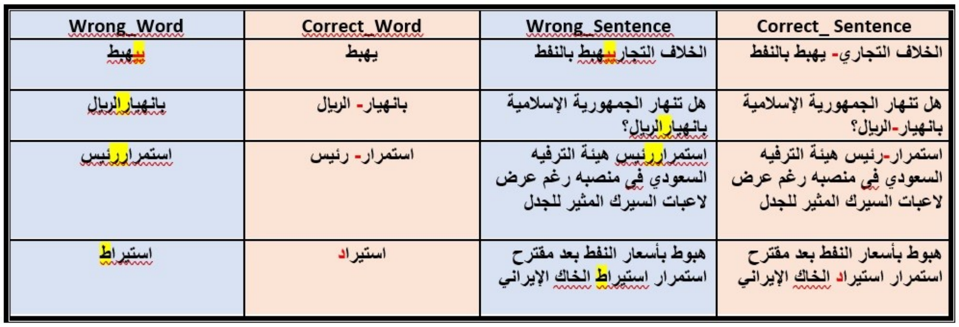
Finally, the output of the LSTM layer used as an input for the output layer which is fully connected layer with only one neuron with a sigmoid activation function to convert the linear input to a probability between 0 and 1. The classification task is carried out using a dense layer with a SoftMax activation function.

4.4.1 Training phase

As shown in Table V-A, the ANS dataset was divided into three sections (training, validation and testing). The proposed CNN-LSTM framework was further evaluated using 80% of data for training, 10% for validation, and 10% for testing.

4.4.2 Testing phase

Several metrics were used to evaluate the proposed model performance (e.g., recall, precision, accuracy, and the F 1- measure), Details can be found in Section V-B. Furthermore, a comparison is made between the proposed model and the works in the literature review to evaluate their performance.



Wrong_Word	Correct_Word	Wrong_Sentence	Correct_ Sentence
ييهبط	يهبط	الخلاف التجاريهبط بالنفط	الخلاف التجاري- يهبط بالنفط
بانهيارالريال	بانهيار- الريال	هل تنهار الجمهورية الإسلامية بانهيارالريال؟	هل تنهار الجمهورية الإسلامية بانهيار-الريال؟
استمراررئيس	استمرار- رئيس	استمراررئيس هيئة الترفيه السعودي في منصبه رغم عرض لاعبات السيرك المثير للجدل	استمرار-رئيس هيئة الترفيه السعودي في منصبه رغم عرض لاعبات السيرك المثير للجدل
استيراط	استيراد	هبوط بأسعار النفط بعد مقترح استمرار استيراط الخاك الإيراني	هبوط بأسعار النفط بعد مقترح استمرار استيراد الخاك الإيراني

Figure3: Examples of Correcting Spellings and Writing Errors.

Algorithm 2 CNN-LSTM algorithm

```
Read dataset, misspelled words, embedding
 dictionary;
foreach word do;
    If word in misspelled words;
     Replace word with correct word;
     Apply word embedding;
     Apply under sampling technique;
     Init max_epochs = 80;
     Init max_acuracy = 0;
     While epoch < max_epochs do;
      Run Training Step;
      Val_loss, val_accuracy = Calculate loss and
accuracy();
foreach validation_ set do;
       If Va_ accuracy > max_accuacy;
  end for
Save the model;
max_accuracy = val_accuracy;
Epoch = epoch + 1;
end for
```

5. EXPERIMENTAL RESULTS OF AFND AND COMPARISONS

This section presents the empirical results of the pro- posed CNN-LSTM using the ANS dataset. The model as evaluated before and after the spelling check. Final results were calculated using the average of all evaluation metrics. Subsection 5. 1 describes the structure of dataset utilized in this study. Subsection 5. 2 presents the hyperparameters values defined by the jSO optimizer. Subsection 5.3 describes the performance measures applied to evaluate the performance of the proposed model. Subsections 5. 4 and 5.5 discuss experimental results with and without jSO.

5.1 Dataset Description

The dataset applied in the current research is called ANS[12]. ANS is a collection of Arabic news headlines. The data was collected from a number of news organizations, including the CNN and BBC [3]. The dataset contains a set of Arabic news headlines, as well as paraphrased and corrupted headlines. It includes two points of view (Claim and instance). The claim dataset has been applied in the current research. A claim dataset consisting of 4547 records was segmented as displayed in Table 1.

Table 1: Dataset Description

Data	Real	Fake	Total
ALL	1475	3152	4547
Training	1035	2150	3185
Validation	150	386	456
Testing	290	616	906

5.2 Optimizing Hyperparameters Values Using jSO Optimization Algorithm

The simulation results have been carried out on the AFNs data utilizing a local machine with processor Intel core i7, 16 GB RAM and NVIDIA GTX 1050i GPU. Furthermore, the programming tasks have been performed using Python 3.7.6, PyTorch 1.8.0, and RMS prop optimizer. Hyperparameters were optimised using the jSO algorithm, which tuned 14 different combinations of 42 possible combinations of each of them. The lower and upper limits of these parameters are shown in Table 2. The minimum and maximum values were carefully chosen to create a network and avoid overfitting. The best validation accuracy was determined by training each of the models generated by the jSO algorithm and evaluating their effectiveness on validation data. According to the evaluation criteria, the optimal values for this model were obtained from the training data set, which increased the accuracy of the validation in training.

5.3 Evaluation Measures

In this paper, the efficiency of the proposed model CNN-LSTM had to be evaluated using standard measures to ensure that the experimental results were statistically significant. To that end, the main performance measures used were accuracy (Acc) [33], which is the number of successful predictions divided by the total number of predictions, which equals the true positive ((TP) + true negative (TN) divided by the TP + TN + false positive (FP) + false negative FN , where TP denotes the number of real records successfully classified by the model.

Table 2: Hyper-parameters Values Utilizing jSO Model.

Hyper-parameters	Min	Max	Optimal
CNN			
Number of Conv-layers	1	3	2
Pool Size of Maxpool	-	2	-
Conv1 Kernal	32	256	26
Conv2 Kernal	32	256	26
Conv3 Kernal	32	256	104
Conv4 Kernal	32	256	-
Numbers of dense layers	0	4	0
dense layer 1	256	1024	989
dense layer 2	256	1024	374
Activation	Leaky RelU	-	
Learning Rate	0.000005	-	0.0003
Batch size	46	-	-
dropout			0.07
LSTM			
Numbers of LSTM layers	1	3	2
Numbers of hidden unit 1	16	86	17
Numbers of hidden unit 2	16	86	41
Numbers of hidden unit 3	16	86	70
Numbers of dense layers	0	4	0
Dense layer 1	256	1024	631
Dense layer 2	256	1024	791
Dropout	-	-	0.24
CNN-LSTM			
Numbers of LSTM layers 1	1	3	1
Numbers of hidden unit	16	86	64
Numbers of hidden unit	16	86	-
Numbers of hidden unit	16	86	-
Number of Conv-layers	1	4	1
Pool Size of Maxpool		2	
Conv1 Kernal	32	256	46
Conv2 Kernal	32	256	
Conv3 Kernal	32	256	
Conv4 Kernal	32	256	
Numbers of dense layers	0	4	0
dense layer 1	256	1024	-
dense layer 2	256	1024	-
dense layer 3	256	1024	-
dense layer 4	256	1024	-
Optimizer Adam	Adam	-	-
Activation Leaky RELU	Leaky	-	-
Learning Rate	0.000005	-	0.000
Batch size	-	-	46

The number of fake records classified as real is denoted by FN , TN denotes the number of fake records classified as fake, and FP denotes the number of real records classified as fake.

Accuracy (Acc) is expressed as in Eq. (11):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

Precision (P) [34], which equals TP divided by $TP+FP$, is expressed as in Eq. (12).

$$P = \frac{TP}{TP+FP} \qquad (12)$$

Recall (R) [35], which equals (TP)divided by $TP+FN$, is expressed as in Eq. (13).

$$R = \frac{TP}{TP+FN} \qquad (13)$$

F -Measure (F_1) [36], which is equal to twice the precision multiplied by recall divided by the sum of precision and recall is written as Eq. (14).

$$F_1 = 2*\frac{Precision * Recall}{Precision + Recall} \qquad (14)$$

5.4 Experimental Results of the Proposed CNN-LSTM

This paper proposes the system that has been requested to deal with the issue of AFND. The proposed CNN-LSTM is utilized for this purpose. The aim is to create a DL model with the best performance in all evaluation metrics.

5.4.1 The results without correcting spelling and errors and Spelling

Figure. 4 presents the accuracy and loss curves of the proposed CNN-LSTM before correcting spelling for training and validation data. Moreover, the proposed model is validated by the evaluation metrics which are explained in the previous subsection. It obtained an accuracy of 67%. Table 3 explains the proposed CNN-LSTM evaluation.

5.4.2 The results after correcting spelling and errors

The proposed model is employed after correcting spelling and errors to demonstrate the effectiveness of the proposed framework. The confusion matrix is visualized in Figure. 5. Moreover, the proposed model is validated by the evaluation metrics after correcting spelling and errors. It obtained an accuracy of 81%.

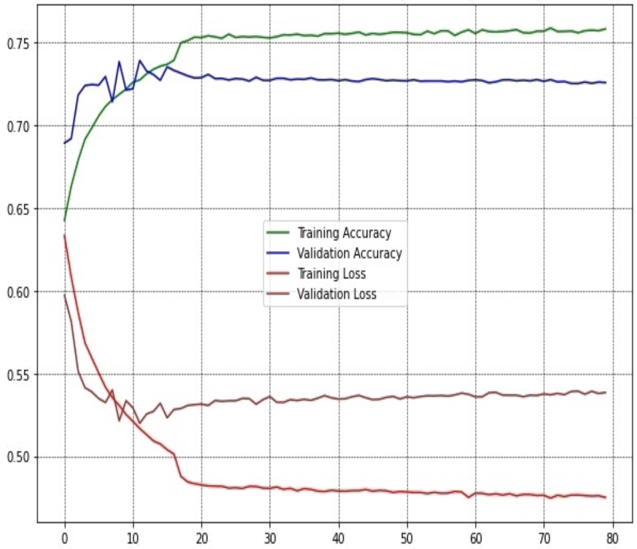
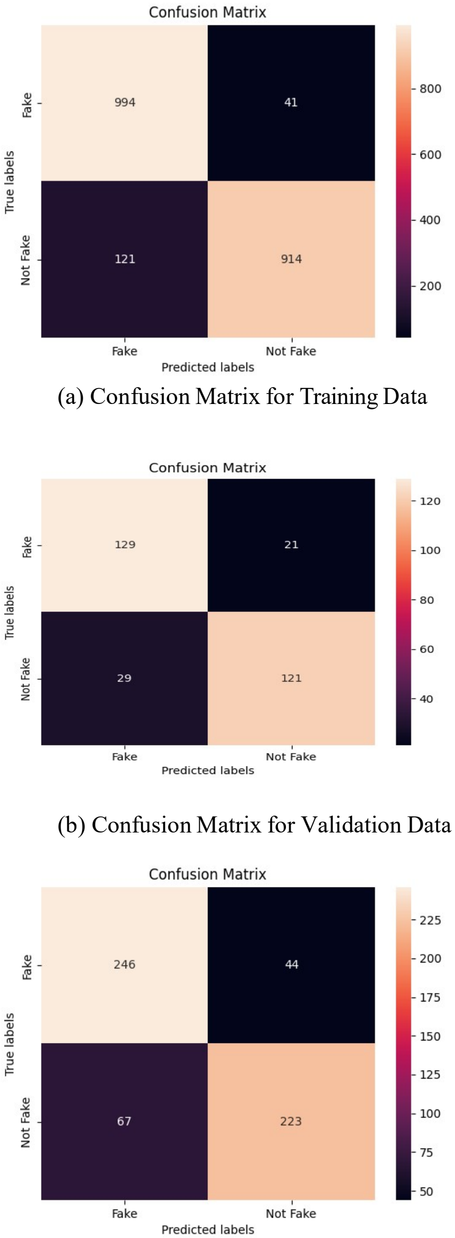


Figure 4: Visual Results Using the Proposed CNN-LSTM Model Before Correcting Spelling.

Table 3: Evaluation Metrics of the Proposed CNN-LSTM Model After Correcting Spelling and Errors

Test	Model	*P*	*R*	*F₁*	*acc*
Training	Fake	0.70	0.37	0.48	
	Real	0.75	0.92	0.83	
	Macro avg	0.73	0.65	0.66	
	weighted	0.74	0.74	0.72	
	Overall	0.95	0.88	0.92	0.74
Validation	Fake	0.59	0.28	0.38	
	Real	0.72	0.91	0.80	
	Macro avg	0.66	0.59	0.59	
	weighted	0.68	0.70	0.66	
	Overall	0.59	0.28	0.38	0.70
Testing	Fake	0.49	0.57	0.52	
	Real	0.78	0.72	0.75	
	Macro avg	0.63	0.64	0.64	
	weighted	0.69	0.67	0.68	
	Overall	0.49	0.57	0.52	0.67

Table 4 explains the proposed CNN-LSTM evaluation after correcting spelling and errors. Finally, Figure 6 shows the accuracy and loss curves of the proposed CNNLSTM after correcting spelling and errors for training and validation data.



(a) Confusion Matrix for Training Data

(b) Confusion Matrix for Validation Data

(c) Confusion Matrix for Testing Data

Figure 5: Confusion Matrix of the Proposed CNN-LSTM Model After Correcting Spelling.

Table 4: Evaluation metrics of the proposed CNN-LSTM model after correcting spelling and errors

Test	Model	*P*	*R*	*F₁*	*acc*
Training	Fake	0.96	0.88	0.92	
	Real	0.89	0.59	0.43	
	Macro avg	0.92	0.92	0.92	
	weighted avg	0.92	0.92	0.92	
	Overall	0.95	0.88	0.92	0.92
Validation	Fake	0.85	0.81	0.83	
	Real	0.82	0.86	0.84	
	Macro avg	0.83	0.83	0.83	
	weighted avg	0.83	0.83	83	
	Overall	0.85	0.81	0.83	0.83
Testing	Fake	0.81	0.80	0.80	
	Real	0.80	0.81	0.81	
	Macro avg	0.81	0.81	0.81	
	weighted avg	0.81	0.81	0.81	
	Overall	0.81	0.81	0.81	0.81

5.4.3 Evaluation metrics of CNN and LSTM both independently after correcting spelling and errors

The results in Tables 5 and 6 reflect the performance of each CNN and LSTM models independently after correcting spelling and errors in terms of P, R, F1, and acc. It is noteworthy that the results of the training, validation and testing data on the four scales were lower than the proposed CNN-LSTM model.

Table 5: Evaluation Metrics of CNN Model After Processing Spelling Data.

Test	Model	*P*	*R*	*F₁*	*acc*
Training	Fake	0.88	0.68	0.77	
	Real	0.75	0.90	0.82	
	Macro avg	0.81	0.80	0.80	
	weighted avg	0.81	0.80	0.80	
	Overall	0.81	0.80	0.80	0.80
Validation	Fake	0.86	0.71	0.78	
	Real	0.75	0.89	0.81	
	Macro avg	0.81	0.80	0.80	
	weighted avg	0.81	0.80	0.80	
	Overall	0.86	0.70	0.77	0.80
Testing	Fake	0.68	0.87	0.76	
	Real	0.82	0.59	0.69	
	Macro avg	0.75	0.73	0.73	
	weighted avg	0.75	0.73	0.73	
	Overall	0.68	0.87	0.76	0.73

Table 6: Evaluation Metrics of LSTM Model After Processing Spelling Data

Test	Model	*P*	*R*	*F₁*	*acc*
Training	Fake	0.86	0.74	0.80	
	Real	0.78	0.88	0.82	
	Macro avg	0.82	0.81	0.81	
	weighted	0.82	0.81	0.81	
	Overall	0.86	0.74	0.80	0.81
	Fake	0.81	0.74	0.77	

Validation	Real	0.76	0.83	0.79	
	Macro avg	0.79	0.78	0.78	
	weighted	0.79	0.78	0.78	
	Overall	0.81	0.74	0.77	0.78
Testing	Fake	0.79	0.79	0.79	
	Real	0.79	0.79	0.79	
	Macro avg	0.79	0.79	0.79	
	weighted	0.79	0.79	0.79	
	Overall	0.79	0.79	0.79	0.79

Finally, Figures 7 and 8 presents the accuracy and loss curves of each CNN and LSTM models independently after correcting spelling for training and validation data.

5.5 Experimental Results with aAnd without jSO Optimizer

To demonstrate the efficiency of the proposed CNN-LSTM, an experiment without and with jSO optimizer is employed. Table 7 displays the extracted results for all performance measures. Table 7 shows that using jSO results in a significant improvement in all performance measures by a sufficient percentage.

5.6 Comparison With the State-of The-Art Models

To measure the proposed model's performance, it is compared to works in the literature, including:

Al-Yahya et al.[3] examined the use of neural networks and three transformer-based language models (AraBERT v02, QARiB, and Ara GPT2). The average overallF1 were 3% ,6%, and 2% for AraBERT v02, QARiB, and Ara GPT2, respectively. Jude Khouja [12] utilized Pre-training of deep bidirectional transformers for language understanding and achieved 64.3%, 65%, and 64% for precision, recall and F1 measure, respectively. In addition, they applied LSTM model. It achieved 34%, 50%, and 40% for precision, recall and F1 measure.

Table 8 presents a comparison between the proposed model and the works stated in the literature on the ANS dataset. As observed from the Table 8, the proposed CNNLSTMobtains the highest score (81) for all evaluation measures to 5 other competing approaches.

Table 7: Evaluation Metrics of the Proposed CNN-LSTM With and Without jSO Optimizer

Model	*P*	*R*	F_1	*acc*
Without jSO	0.79	0.73	0.76	0.77
With jSO	0.81	0.81	0.81	0.81

Table 8: Comparison Between the Proposed Model and Works in the Literature.

Model	*P*	*R*	F_1	*acc*
AraBERT v02 [3]	0.06	0.01	0.03	0.68
QARiB [3]	0.11	0.05	0.06	0.67
AraGPT2 [3]	0.27	0.19	0.20	0.64
LSTM [12]	0.34	0.50	0.40	0.67
BERT [12]	0.64	0.65	0.64	-
Proposed LSTM-CNN before correcting mistakes	0.69	0.67	0.68	0.67
Proposed LSTM-CNN after correcting mistakes	**0.81**	**0.81**	**0.81**	**0.81**

6. CONCLUSION

This study detected FN using the Arabic language. The Arabic language is known to be very difficult in terms of analysis. The current study proposed an AFND system based on a hybrid DL model that includes both CNN and CNN-LSTM modalities. It applied the proposed CNN-LSTM to the ANS group dataset. The proposed model was employed on the claim data with and without addressing the spelling problem. The experimental results indicated that the proposed approach provides significant results in AFND after addressing the spelling problem rather than using the dataset without correcting the spelling. The proposed CNN-LSTM achieved an accuracy, precision, recall, and F1-measure of 81%.

Moreover, the comparison with CNN, LSTM, and recent models show that the proposed models have superior performance among the investigated techniques with regard to the issue of AFND. Therefore, the proposed models can be considered an efficient AFND system. In the future, we aim to apply different Arabic datasets and compare the impact of different embedding methods on the performance of classifiers to achieve the highest accuracy results. We would also like to see some clear and specific techniques for assessing the quality of the embeddings.

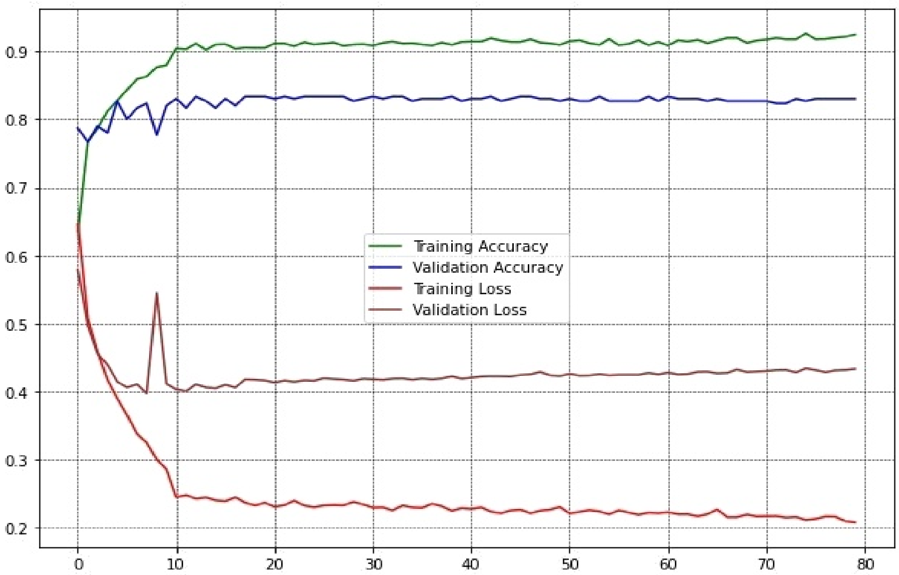


Figure 6: Visual results of the proposed LSTM-CNN after correcting spelling

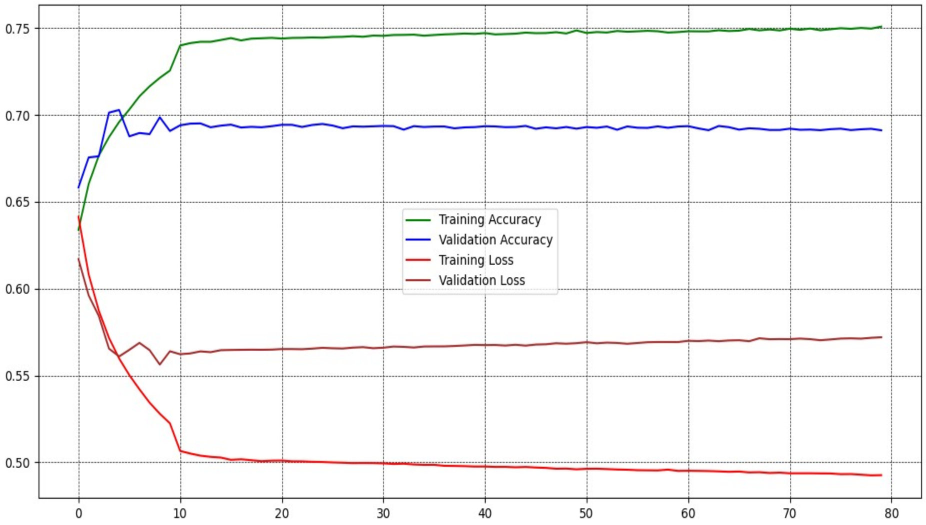


Figure 7: Visual results of CNN model after correcting spelling.

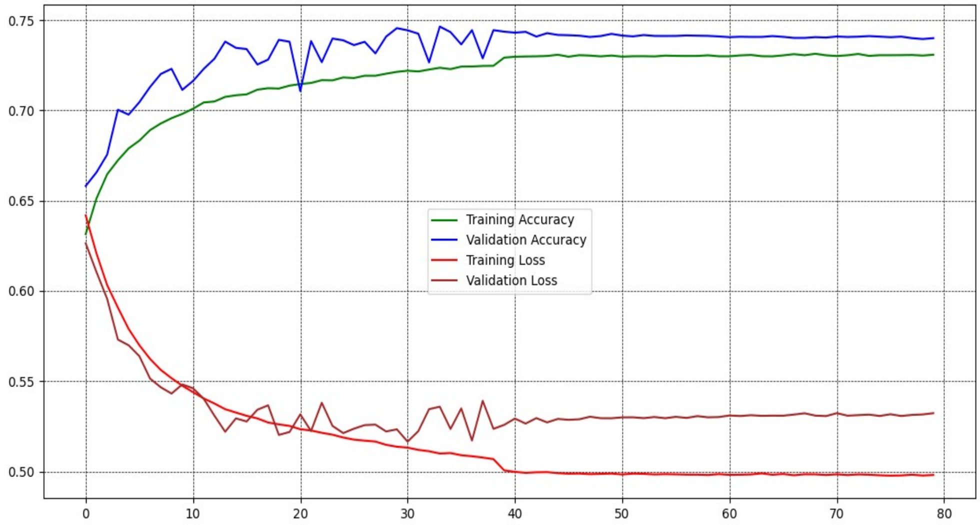


Figure 8: Visual results of LSTM model after correcting spelling.

REFERENCES

[1] T. Nygren, M. Guath, C.-A. W. Axelsson, and D. Frau-Meigs, "Combatting visual fake news with a professional fact-checking tool in education in France, Romania, Spain and Sweden," *Information,* vol. 12, no. 5, p. 201, 2021.

[2] J. Zeng, Y. Zhang, X. Ma, and Society, "Fake news detection for epidemic emergencies via deep correlations between text and images," *Sustainable Cities,* vol. 66, p. 102652, 2021.

[3] 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.

[4] P. Patwa *et al.*, "Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts," in *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, 2021, pp. 42-53: Springer.

[5] S. Admoko, N. Suprapto, U. Deta, H. Achmadi, and E. Hariyono, "Using Toulmin's Argument Pattern Approach to Identify Infodemics in the Covid-19 Pandemic Era," in *Journal of Physics: Conference Series*, 2021, vol. 1805, no. 1, p. 012011: IOP Publishing.

[6] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems,* vol. 117, pp. 47-58, 2021.

[7] R. F. Rasmus Kleis Nielsen, Nic Newman, J Scott Brennen, and a. P. N. Howard, "avigating the 'infodemic': How people in six countries access and rate news and information about coronavirus," 2020. Reuters Institute

[8] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing,* vol. 100, p. 106983, 2021.

[9] M. Kolhar, F. Al-Turjman, A. Alameen, and M. M. Abualhaj, "A three layered decentralized IoT biometric architecture for city lockdown during COVID-19 outbreak," *Ieee Access,* vol. 8, pp. 163608-163617, 2020.

[10] S.-H. Wang, Y.-D. Zhang, M. Yang, B. Liu, J. Ramirez, and J. M. Gorriz, "Unilateral sensorineural hearing loss identification based on double-density dual-tree complex wavelet transform and multinomial logistic regression," *Integrated Computer-Aided Engineering,* vol. 26, no. 4, pp. 411-426, 2019.

[11] Y.-J. Lu and C.-T. Li, "GCAN: Graph-aware co-attention networks for explainable fake news detection on social media," *arXiv preprint arXiv:.11648,* 2020.

[12] J. Khouja, "Stance prediction and claim verification: An Arabic perspective," *arXiv preprint arXiv:.10410,* 2020.

[13] S. Ilić, E. Marrese-Taylor, J. A. Balazs, and Y. Matsuo, "Deep contextualized word representations for detecting sarcasm and irony," *arXiv preprint arXiv:.09795,* 2018.

[14] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *International Journal of*

Information Management Data Insights, vol. 1, no. 1, p. 100007, 2021.

[15] N. Rani, P. Das, A. K. J. C. Bhardwaj, C. Practice, and Experience, "Rumor, misinformation among web: A contemporary review of rumor detection techniques during different web waves," vol. 34, no. 1, p. e6479, 2022.

[16] F. Papanastasiou, G. Katsimpras, and G. Paliouras, "Tensor factorization with label information for fake news detection," *arXiv preprint arXiv:.03957,* 2019.

[17] R. K. Kaliyar, A. Goswami, and P. Narang, "DeepFakE: improving fake news detection using tensor decomposition-based deep neural network," *The Journal of Supercomputing,* vol. 77, no. 2, pp. 1015-1037, 2021.

[18] P. Qi, J. Cao, T. Yang, J. Guo, and J. Li, "Exploiting multi-domain visual information for fake news detection," in *2019 IEEE International Conference on Data Mining (ICDM)*, 2019, pp. 518-527: IEEE.

[19] A. P. B. Veyseh, M. T. Thai, T. H. Nguyen, and D. Dou, "Rumor detection in social networks via deep contextual modeling," in *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019, pp. 113-120.

[20] J. Ma, W. Gao, and K.-F. Wong, "Detect rumor and stance jointly by neural multi-task learning," in *Companion proceedings of the the web conference 2018*, 2018, pp. 585-593.

[21] B. Ghanem, S. P. Ponzetto, P. Rosso, and F. Rangel, "Fakeflow: Fake news detection by modeling the flow of affective information," *arXiv preprint arXiv:.09810,* 2021.

[22] J.-S. Shim, Y. Lee, and H. J. E. S. w. A. Ahn, "A link2vec-based fake news detection model using web search results," vol. 184, p. 115491, 2021.

[23] V. Jain, R. K. Kaliyar, A. Goswami, P. Narang, and Y. Sharma, "AENeT: an attention-enabled neural architecture for fake news detection using contextual features," *Neural Computing Applications,* vol. 34, no. 1, pp. 771-782, 2022.

[24] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging,* vol. 9, no. 4, pp. 611-629, 2018.

[25] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with ReLU activation function and linear spline-type methods," *Neural Networks,* vol. 110, pp. 232-242, 2019.

[26] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems,* vol. 22, no. 7, pp. 4507-4518, 2020.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[28] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation,* vol. 12, no. 10, pp. 2451-2471, 2000.

[29] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jSO," in *2017 IEEE congress on evolutionary computation (CEC)*, 2017, pp. 1311-1318: IEEE.

[30] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE congress on evolutionary computation (CEC)*, 2014, pp. 1658-1665: IEEE.

[31] A. A. Altowayan and L. Tao, "Word embeddings for Arabic sentiment analysis," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 3820-3825: IEEE.

[32] W. Nugraha, M. S. Maulana, and A. Sasongko, "Clustering Based Undersampling for Handling Class Imbalance in C4. 5 Classification Algorithm," in *Journal of Physics: Conference Series*, 2020, vol. 1641, no. 1, p. 012014: IOP Publishing.

[33] M. Yin, J. Wortman Vaughan, and H. Wallach, "Understanding the effect of accuracy on trust in machine learning models," in *Proceedings of the 2019 chi conference on human factors in computing systems*, 2019, pp. 1-12.

[34] A. K. A. d. Medeiros *et al.*, "Process mining based on clustering: A quest for precision," in *International conference on business process management*, 2007, pp. 17-29: Springer.

[35] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval,* vol. 12, no. 4, pp. 461-486, 2009.

[36] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "Combining evaluation metrics via the unanimous improvement ratio and its application to clustering tasks," *Journal of Artificial Intelligence Research,* vol. 42, pp. 689-718, 2011.