E-ISSN: 2676-6167 | P-ISSN: 2538-5100

Journal of Applied Research on Industrial Engineering

www.journal-aprie.com

J. Appl. Res. Ind. Eng. Vol. 8, Spec. Issue. (2021) 1–17.

Paper Type: Research Paper

A Decision-Making System for Detecting Fake Persian News by Improving Deep Learning Algorithms– Case Study of Covid-19 News

Vahid Mottaghi[1], Mahdi Esmaeili[2,*], Ghasem Ali Bazaee[3], Mohammadali Afshar Kazemi[3]

[1] Department of IT Management, Qeshm Branch, Islamic Azad University, Qeshm, Iran; mvahid500@gmail.com.

[2] Department of Computer Science, Kashan Branch, Islamic Azad University, Kashan, Iran; m.esmaeili@iaukashan.ac.ir.

3 Department of Management, Central Tehran Branch, Islamic Azad University, Tehran, Iran; gh.bazaei@iauctb.ac.ir; academy.tehran1@gmail.com.

Citation:

Mottaghi, V., Esmaeili, M., Bazaee, Gh. A., Afshar Kazemi, M. (2021). A decision-making system for detecting fake persian news by improving deep learning algorithms– case study of Covid-19 news. *Journal of applied research on industrial engineering, 8* (Spec. Issue), 1-17.

Received: 14/04/2021 Reviewed: 01/06/2021 Revised: 18/07/2021 Accepted: 21/08/2021

Abstract

With the increase of news on social networks, a way to identify fake news has become an essential matter. Classification is a fundamental task in Natural Language Processing (NLP). Convolutional Neural Network (CNN), as a popular deep learning model, has shown remarkable success in the task of fake news classification. In this paper, new baseline models were studied for fake news classification using CNN. In these models, documents are fed to the network as a 3-dimensional tensor representation to provide sentence-level analysis. Applying such a method enables the models to take advantage of the positional information of the sentences in the texts. Besides, analyzing adjacent sentences allows extracting additional features. The proposed models were compared with the state-of-the-art models using a collection of real and fake news extracted from Twitter about covid-19, and the fusion layer was used as the decision layer in selecting the best feature. The results showed that the proposed models had better performance, particularly in these documents, and the results were obtained with 97.33% accuracy for classification on Covid-19 after reviewing the evaluation criteria of the proposed decision system model.

Keywords: Fake news, Text classification, Decision-making, Deep learning, Convolutional neural network, Natural language processing.

1 | Introduction

Recent political events have led to the popularity and spread of fake news. Some of this news spreads in a way that is very close to reality and affects many people. A variety of methods have been proposed to automate the process of fake news detection, the most popular of which are blacklists “from unreliable sources and authors”. While these tools are useful, we should consider more items to create a more complete solution for such news. Machine learning methods and Natural Language Processing (NLP) techniques are useful tools for detecting fake news [1]-[4].

Corresponding Author: m.esmaeili@iaukashan.ac.ir

http://dx.doi.org/10.22105/jarie.2021.281257.1299

Even for human beings, it is very difficult to separate fake news from real news. Like other areas of research, the field of fake news has several challenges, the most important of which are the followings:

- *Linguistic complexity: The use of language in fake news is complex. In [5], the difference between using fake news language and real news language were analyzed in three types of fake news (humor, deception, and advertising). It showed that a wide range of linguistic factors is involved in shaping fake news. According to this research, using feature-based approaches are costly and time-consuming.*
- *There is no boundary between fake and real news: Using false details, fake news often make real stories confusing, which are difficult to be distinguished correctly. Often a fake news maker combines a true story with false details to mislead people [6].*
- *Lack of tagged data: Fake news information is limited. Currently, only fake political news databases have been published. Non-political areas are still open for future research [1].*

Many of the approaches presented in the literature for detecting fake news consider this issue as a problem in text classification. Their purpose is to associate tags such as fake or real, true or false, with a particular text. Text classification is a fundamental task in NLP. The task is the process of assigning a class label, from a set of predefined classes, to a given text according to its content, which has many applications such as sentiment analysis [7], spam detection [8], and topic categorization [9].

Text classification can be done manually or automatically. Although the manual method is more accurate, it is very costly and time-consuming. Therefore, to provide scalability, several machine learning, NLP, and other techniques are used for automatic text classification. Supervised learning is a machine learning task of learning a function (classifier) using prelabeled samples as a training dataset [10]. A key step in supervised learning is feature extraction. Traditional machine learning methods represent the text with handcrafted methods, e.g., n-grams [11]. Recently, deep learning methods have been used for automatic feature extraction, including Convolutional Neural Networks (CNNs) [12], Recurrent Neural Networks (RNNs) [13], and particularly Long Short-Term Memory (LSTM) [14]. In this paper, we present a new baseline model for fake news classification using CNN. In this model, documents are fed to the network as a 3-dimensional tensor representation to provide sentence-level analysis.

The main contributions of this research can be summarized as follows:

- *A new representation of texts in a 3-dimensional tensor.*
- *Presenting a sentence-level CNN.*
- *Considering positional information of sentences for text classification.*
- *Analyzing adjacent sentences for extracting additional features.*
- *Finally, the result of the number of fake and real news at the sentence level is obtained based on the evaluation criteria of the introduced algorithm. This has the advantage of not losing useful information during the analysis compared to other methods.*

Feature selection methods for dealing with high-dimensional data: choosing the right feature can improve the learning process. In-depth learning features are extracted automatically. In general, in deep learning methods, two steps of feature extraction and classification are performed by the model and hidden layers of the neural network. The aim of using neural networks is to identify patterns and make simple decisions about them. Convolutional networks are also a type of neural network and their architecture is a list of layers. CNNs use three basic ideas: local receptive fields, shared weights, and pooling.

Mottaghi et al.| J. Appl. Res. Ind. Eng. 8(Spec. Issue) (2021) 1-17

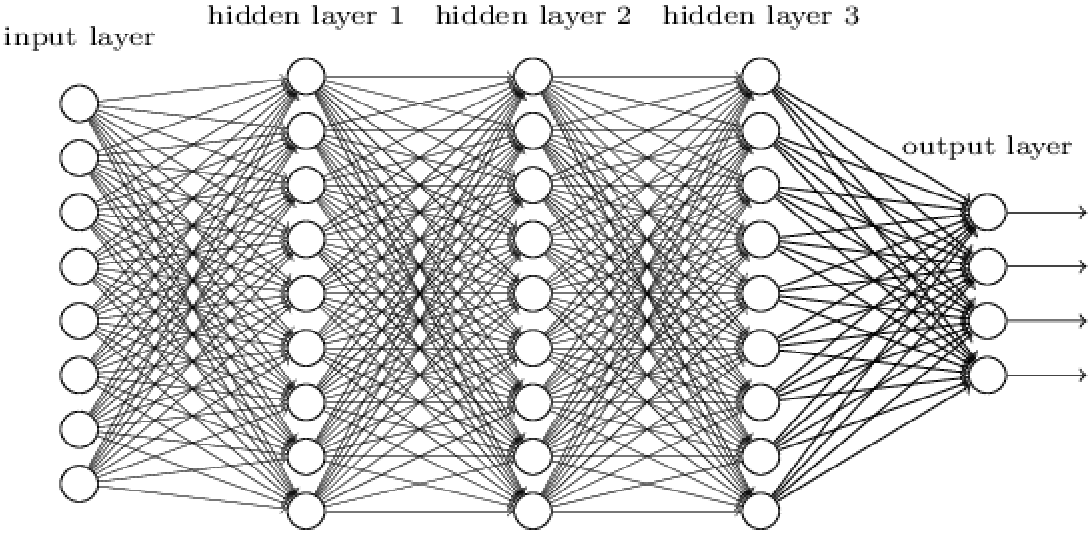

Fig. 1. Shape of the Convolutional neural network.

This paper is structured as follows: the previous works are summarized in the next section. The details of the proposed methods are described in Section 3. We evaluated our approach on the covid-19 dataset. The experimental results are presented in Section 4. Finally, the paper concludes with future research directions in Section 5.

2 | Related Works

As mentioned, fake news is a subset of the problems of text classification that aims to assign tags, such as fake or real, to a particular text. In most cases, researchers have used machine learning and deep learning approaches, and have achieved promising results. On the other hand, some researchers have used other approaches based on data mining techniques, such as time series analysis, and have used external sources (e.g., knowledge base) to predict the class of documents. Therefore, we first deal with text classification techniques and then refer to the approaches in the literature for detecting fake news.

2.1 | Text Classification Approaches

Different approaches have been proposed for text classification. Initial approaches were based on the classical machine learning techniques, which followed two stages, i.e., extracting handcrafted features and classifying the documents. Typical features include Bag-of-Words (BOW), n-grams, and their Term Frequency-Inverse Document Frequency (TF-IDF) [15]. To improve the performance of these approaches, some strategies such as term weighting can be used to assign appropriate values to each term [16]. Alternatively, several recent studies have shown the success of deep learning on text classification. As the neural networks receive their inputs numerically, word embeddings e.g., word2vec [17] or global vectors (Glove) [18] are usually used to represent words as numerical vectors by capturing the similarities/regularities between words.

There are a variety of deep learning models for text classification. Due to the sequential nature of textual data, RNNs, including LSTM and Gated Recurrent Units (GRU) [19], have been widely used in text processing. For example, in [20] research, authors examined generative and discriminative LSTM models for text classification. They found that although the generative models perform better than BOW, they have higher asymptotic error rates than discriminative RNN-based models. Another popular model is CNN, which was originally invented for computer vision [21]. Subsequently, CNN models were applied in NLP and achieved excellent results [22]. Many researchers have worked on the effective use of CNNs in text classification, since a single-layer, word-level CNN was successfully used in sentence classification with pretrained word embeddings [23]. The proposed method in [15] was the first attempt to perform text classification entirely at the character level and reported competitive results. Their models use 70 characters by one-hot encoding, including 26 English letters, 10 digits, 33 other characters, and the new line character. Conneau et al. [24] adopted very deep convolutional networks, i.e., residual neural network (ResNet) [25], to the character-level (char-level) text classification.

Some researchers tried to improve the performance of the models by applying extra mechanisms. Attention is one of the most effective mechanisms that select significant information to achieve superior results [26]. Deep Neural Networks (DNN) with attention mechanisms can yield better results. Some of the remarkable examples include source-target attention and self-attention [27]. Particularly, a two-level attention mechanism, including word attention and sentence attention, was developed on GRU by [28] for document classification. In [29] research, authors used dense connections with multi-scale feature attention to produce variable n-gram features. Since the present paper aimed to present a new baseline model, employing such mechanisms was avoided.

2.2 | Fake News Classification Approaches

Most fake news detection approaches focus on using categorical content features. According to [2], very few fake news detecting-approaches have relied on purely social models. Most researchers have to test different classification algorithms to find the most suitable model for their dataset. These approaches can be incorporated into classification and other approaches. Classification approaches can in turn be based on machine learning and deep learning.

2.2.1 | Machine learning approaches

Machine learning algorithms are useful for solving many problems in the field of information engineering. The first approaches focused on social networks credibility [30], while the discovery of deceptions by computers has yielded promising results [31]. Machine learning techniques have been tested and evaluated in many studies on the issue of misinformation. Most machine learning approaches used for fake news and rumor detection use a supervised learning strategy. Here are some of these algorithms and attempts to detect fake news.

- *Support Vector Machine (SVM): It is one of the most common classification methods in some research fields. SVMs are discriminatory classifications that are formally defined by a delimiter and try to find the boundary that has the most secure margin between the datasets. According to experiments in [31], SVMs outperformed many supervised machine learning approaches in detecting deceptions in texts, scoring F1 of 0.84. However, as noted by the authors, depending on the dataset selected for training, there can be significant differences in its performance [31].*
- *Decision Tree (DT): Another family of algorithms that has been extensively studied, especially for rumor analysis tasks, is the DT [32]. The DT performs a recursive division of attribute values to determine the class. The DT for data is generated by algorithms such as J48 (C4.5) [33]. Despite their relative simplicity over other machine learning schemes, they have achieved competitive performance in many tasks. The effectiveness of the J48 DT comparing other algorithms, including SVM, is shown in [34] and [35]. In [35] research, the authors proposed a set of user confidence metrics to assess the reliability of users on social networks through the DT and reached an accuracy of 0.75.*
- *Logistic Regression (LR): Learning algorithms based on LR models have also been used in several studies, specifically for rumor classification tasks. LR has achieved acceptable performance compared to DT and SVM in the field of fake news [36], [4], [32].*
- *Hidden Markov Models (HMM): It is able to control sequence-based data. HMMs were exploited to classify rumors in [37].*

2.2.2 | Deep learning approaches

In recent years, with the increasing power of Graphical Processing Units (GPUs) and the availability of a massive amount of data, deep learning techniques have achieved many successes. Along with that,

Mottaghi et al.| J. Appl. Res. Ind. Eng. 8(Spec. Issue) (2021) 1-17

NLP has changed. In many NLP tasks, deep learning gained a great deal of efficiency over the traditional machine learning and statistic techniques that were used a few years before [38]. Deep learning is very beneficial in text generation, vector representation, word representation estimation, sentence classification, SA, sentence modeling, and feature representation [39]. One of the main advantages of these techniques is that they do not require manually-tuned features based on expert knowledge and available linguistic resources [40]. But one of the limitations of neural networks when working with text data is that raw texts cannot be given to the networks, because neural networks receive the data as a vector and generate outputs as a vector. Instead of using the unique dimensions for every feature, they try to embed each feature into a d-dimensional space and represent it as a dense vector in the space. The most important advantage of these vectors is that similar words fall into the vector space closely [41]. Deep learning includes various types of Artificial Neural Networks (ANNs) such as CNNs, RNNs, LSTM, GRUs, and CapsuleNet, which we continue to focus on each of them individually.

Convolutional Neural Networks (CNNs): CNNs are a kind of DNN initially introduced by [42] for image recognition. Today, these networks are used for various tasks such as face recognition [43], human pose estimation [44], speech recognition [45], and NLP. CNN is a kind of feedforward neural network with features such as convolution layers, a sparse connection, parameter sharing, and pooling [40]. These networks consist of three main layers. i) Convolution layer: In these layers, CNN attempts to extract features by using kernels on the input feature map or intermediate feature map. The advantage of these layers is sharing of parameters, which reduces network parameters severely. ii) Pooling layer: These layers are used to reduce network parameters and prevent overfitting. Max pooling and average pooling are the most commonly used pooling strategies. Among the layers of CNN, pooling has been most studied, and different strategies have been proposed for it, such as stochastic pooling [46], spatial pyramid pooling [47], and def pooling [48]. Assuming having a 8*8 feature map, a 3*3 kernel, and with stride 2, the output of the input feature map reduced to a feature map 4*4. In max pooling, the highest amount of input feature map that the kernel is applied to, and in average pooling, the average features are selected [49]. iii) Fully connected layers: They are similar to traditional neural networks and form 90% of CNNs parameters. The task of these layers is to convert 2-dimensional feature map into a 1- dimensional feature vector to calculate the score of the categories or to continue the learning process. The disadvantage of these layers is the presence of a lot of parameters in them, which is a great deal to learn [49]. CNNs in fake news, have also achieved a lot of success. [50], [51], [2] researches are examples of works that have used these networks.

Recurrent Neural Networks (RNN): RNNs are feedforward networks that add the concept of time to the model. This concept is defined by the edges in adjacent steps. Edges that connect adjacent times are called 'recurrent edge'. These edges may create cycles. Among the cycles that can be created, a cycle is of length 1, which indicates the connection to itself over time. In these networks, the status at any moment depends on the current input and the previous step [13]. If the current input is X_t and the previous network state is h_t, which is taken from the network hidden node, the output is calculated by the following equations:

$$y^{t} = \text{softmax}\left(w^{hy}\, h_t + b_y\right). \tag{1}$$

h^t is obtained from the following equation:

$$h^{t} = (w^{hx} * x_t + w^{hh} * h_{(t-1)} + b_n. \tag{2}$$

In these equations, w^{hx}, w^{hh}, and w^{hx} are weight matrix that can be learned, while b_n and b_y are bias values, which allow each node to learn an offset [13]. In these networks, during the backpropagation at long time steps, the vanishing and exploding gradients occur. LSTM network was offered as a solution to these problems by [14]. These networks are similar to RNN networks, except that each node in the hidden layer is replaced by a memory cell. Each memory cell contains a node with a self-connected recurrent edge and assures that the gradient can pass through many time steps without vanishing and exploding gradients [14]. The LSTM networks include four gates: i) input gate, ii) output gate, iii) update gate, and iv) forget gate [52]. Another kind of recurrent networks that is proposed to deal with the vanishing and exploding gradient problems are GRU networks. These networks were first designed by Kyungnyuncho for natural machine translation [53]. Given the structure of these networks and the nature of natural language, they are a good choice for fake news classification. [54], [35], [55] researches are examples of works that have used these networks.

CapsuleNet CNNs work well for classification tasks. These networks work best when classification data is similar to train data. In image data, if rotations or translations occur on the image, the performance of these networks becomes very bad [56]. CNNs use the pooling layer to reduce the size of the input features map or the intermediate features map. On the other hand, pooling is also affected in preventing overfitting. Max pooling selects the most active neuron between several neurons in the low layer and ignores the rest, which causes the spatial connections to be removed and the space between the features be ignored [57]. CapsuleNet was introduced by [58] and [57] to resolve the problems of CNNs. In these networks, the output is a vector instead of a scalar value, indicating the existence of an entity. This vector contains all the attributes of an entity and the length of this vector is the probability of existence. Each of the low-layer capsules sends its outputs to higher-level capsules. Using vector four of the capsules output to the dynamic routing mechanism allows the capsules output to be sent to the parent capsule more appropriately. At first, the capsule output may be sent to all the capsules in the parent's layer, but over time, the dynamic routing algorithm is sent to the optimal output by setting the coupling coefficient [57]. Hinton et al., in [58] introduced and improved the version of the CapsuleNet. In these networks, each capsule uses a logical unit to represent the presence of an entity and uses a matrix to describe its properties. Accordingly, the matrix is called the 'Matrix POS', which is a 4*4 matrix, and is used to represent various features such as spatial coordinates, size, and other characteristics of a feature. In this approach, they used Expectation-Maximization (EM) for routing. The purpose of EM routing is to classify capsules to create a complete relationship using the EM clustering method [58]. The main advantage of CapsuleNet over CNNs is that they require less training set and learn the good view of each data class than CNNs [57]. CapsuleNet has achieved better results over many issues than CNNs. For example, in [57], CapsuleNet arrived at most datasets at a 25% error rate, which was 14% less error than the baseline model. [59]-[62] researches are other examples of using CapsuleNet for fake news classification.

Other machine learning algorithms such as optimization algorithms [63], fuzzy algorithms [64] and [65], heuristic [66] and [67], and metaheuristic algorithms can be used in the literature in the composition of those models [68]-[71].

3 | Method

In this section, we describe the architecture of the proposed Sentence-Level Convolutional Neural Network (SLCNN) for classifying fake news documents. The key idea of the model is that using the positional information of each sentence in the document may improve the performance of the classifier. Furthermore, analyzing adjacent sentences allows extracting some extra features, e.g., writing style

Features, which can be useful in some applications, such as spam review detection and fake news detection. Hence, we present two baseline models for text classification task, based on the CNN architecture. For this purpose, we introduce a 3-dimensional representation of the documents to enable sentence-level analysis. The preprocessing phase, the architecture of the SLCNN, and its variant SLCNN+V are explained in the following subsections.

Mottaghi et al.| J. Appl. Res. Ind. Eng. 8(Spec. Issue) (2021) 1-17

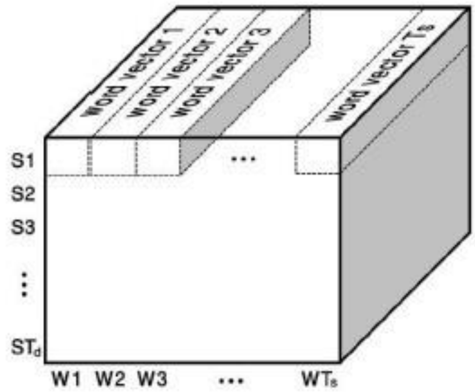

Fig. 2. Shape of the converted documents.

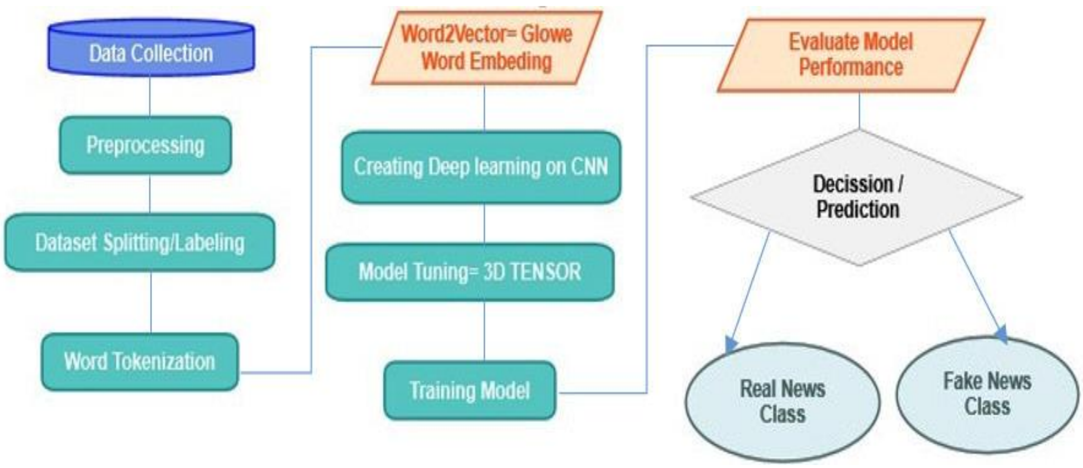

Fig. 3. Proposed framework for detecting fake news at sentence level based on 3D tensor-based CNN.

3.1 | Preprocessing

During the preprocessing phase, the documents are cleaned by removing some unimportant characters, like the HTML tags and the punctuations. Then all words are normalized by converting to their lowercase forms. After that, as the most important step, each document is transformed into a 3-dimensional tensor, illustrated in *Fig. 1*. As shown in the figure, the sentences of the document form the first dimension of the tensor. In the same way, the words of the sentences shape the second dimension, while the third dimension represents the word vectors of the words. The pretrained word embeddings, e.g., word2vec and Glove, could be used for representing the word vectors.

Since the input size of the network must be fixed, and due to the different sizes of both the texts and the sentences, we considered two thresholds, one for the number of sentences in the documents, T_d, and the other for the number of words in the sentences, T_s. The documents and the sentences longer than the thresholds would be cropped and shorter ones would be padded by zeros.

After some statistical analysis on the datasets in our experiments, as well as considering the structure of the SLCNN, we chose $T_s = 46$. The threshold for the number of sentences in the documents is calculated by the following equation:

$$T_d = \lceil \mu + 1.5\sigma \rceil. \qquad (3)$$

In this equation, μ is the average number of sentences in the documents, and σ is the standard deviation. As a result, the outlier sizes are ignored to prevent the model from constructing very large and sparse tensors. The relevant statistical data is provided in Section 4.

3.2 | Architecture

The architecture of the proposed models is illustrated in *Fig. 2*. Overall, in the input layer, the documents are provided in the form of the 3D tensor, introduced in Section 3.1. After that, using four Horizontal Convolutional Blocks (HCB), one feature per filter is extracted for each sentence individually. In other words, one feature vector for each sentence is provided, just before the fully connected layers, with the size equal to the number of filters. In this way, in addition to the word-level features, the positional information of the sentences is also used in the learning process. Moreover, as mentioned before, analyzing adjacent sentences can extract some useful features. For this purpose, the second model (SLCNN+V) is created by adding a Vertical Convolutional Block (VCB) before fully connected layers. Finally, two fully connected (dense) layers end to the output layer.

Looking at the details of the convolutional blocks, as shown in *Fig. 3*, there are two sequential convolution layers, each one followed by a Rectified Linear Unit (ReLU) activation function, $f(x) = max\ (0,x)$. A convolution operation consists of a filter $w \in R^{s*t*d}$, which is applied to each possible window of $s*t$ features from its input feature map, X, to produce a new feature map by *Eq. (5)*:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,1} & \cdots & x_{2,n} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{m,1} & x_{m,1} & \cdots & x_{m,n} \end{bmatrix}. \tag{4}$$

$$x'_{i,j} = f\left(w.x_{i,j:i\ +\ s-1,j+t-1} + b\right). \tag{5}$$

In these equations, $x_{i,j:y,z}$ is the concatenation of features within the specified interval, $b \in R$ is a bias term, and f is a non-linear function such as the ReLU. For the HCB, we consider s = 1 and t = 2, while for the VCB, s = 2 and t = 1. It should be noted that in the first convolution layer of the first HCB, d (the third dimension of the filters) is equal to the size of the word vectors, and in other cases d = 1. At the end of the blocks, there is a max pooling operation, with the pooling size = 2, that is applied over the generated intermediate feature map to select the maximum value from any two adjacent features as a more important feature. The new feature map is calculated by the following equation:

$$x'_{i,j} = \begin{cases} \max\left\{x_{i,2j-1}\ , x_{i,2j}\right\} & \text{, for the HCB} \\ \max\left\{x_{2i-1,j}, x_{2i,j}\right\} & \text{, for the VCB} \end{cases}. \tag{6}$$

The fusion layer, as a decision-making layer, tries to select important features from the two received branches. An early fusion layer is used in the proposed architecture. This layer decides the features according to the weights learned during the backpropagation time.

The process of extracting one feature from one filter is described as: the model uses multiple filters to obtain multiple features. The final extracted features are passed to the fully connected layers that end to a softmax output layer, which is the probability distribution over labels. For regularization, a dropout module [58] is employed after each fully connected layer.

Mottaghi et al.| J. Appl. Res. Ind. Eng. 8(Spec. Issue) (2021) 1-17

4 | Experiments

4.1 | Experimental Settings

The Natural Language Toolkit (NLTK) was used to tokenize words and sentences. In the input layer, as mentioned before, pretrained word embeddings were used to convert the words into the corresponding word vectors. We used 100-dimensional Word2vec in our experiments. Out-of-Vocabulary (OOV) words were initialized from a uniform distribution with a range of [-0.01, 0.01]. We set the number of filters to 128 for all the convolutional blocks. Also, we considered two different sizes for fully connected layers, shown in *Table 1*. Both the dropout rates were set to 0.5. The model's parameters were trained by the Adam optimizer [72], with an initial learning rate of 0.001. The model was implemented using Keras and was run for 50 epochs.

Table 1. Fully connected layers in this experiment.

Layers	Small	Large
Fully-connected 1	512	1024
Fully-connected 2	512	1024
Output	Depends on the problem	

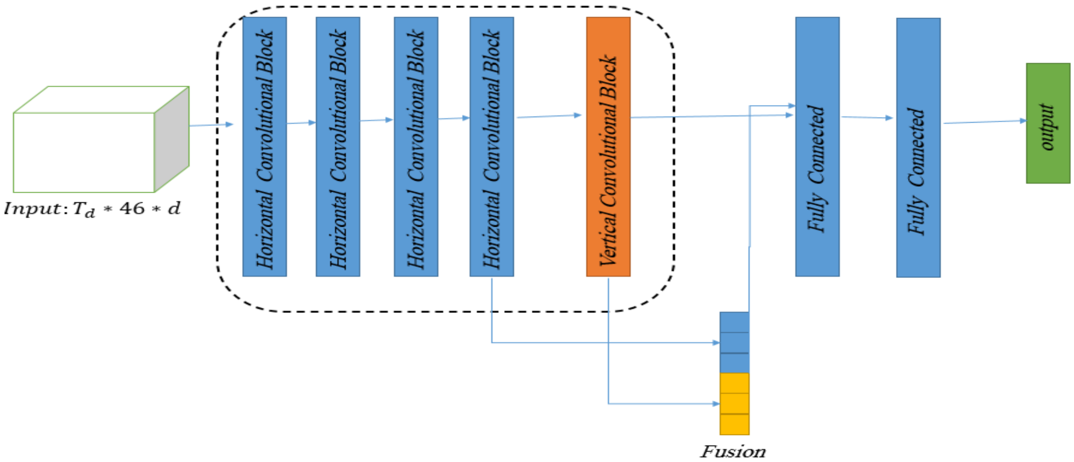

Fig. 4. Architecture of the proposed models. The dashed block (VCB) is used only in SLCNN+V.

Table 2. Datasets in our experiments.

Datasets	AG news	DBpedia	Yelp. P	Yelp. F	Amazon. P	Amazon. F
# of training samples	120k	560k	560k	650k	3600k	3000k
# of test samples	7.6k	70k	38k	50k	400k	650k
# of classes	4	14	2	5	2	5

4.2 | Covid-19 Dataset

The data of this research was collected for 2 months, from 20/03/2020 to 25/05/2020. This data was used for training and testing the model. This data was collected and completed in several steps. In each step, more information was added to the original data. The data was stored in separate rows; each row could be considered corresponding to a user and input data for model training. Next, we started by collecting the initial data and then, in each step, we checked the completion of the initial data and the addition of the required features. Finally, the statistical information of all the collected data was presented.

Table 3. The statistical information of the datasets.

Statistics	AG News	DBpedia	Yelp. P	Yelp. F	Amazon. P	Amazon. F	Covid-19
# of sentences	164k	1505k	5082k	5958k	18654k	16986k	69k
Cropped sentences (%)	2	2.9	2.6	2.6	2.4	2.5	2.2
Cropped documents (%)	0.4	3.6	6	6	3.1	3	2.3
Documents that contain	2.5	6.9	16.1	16.4	10.3	10.9	2.5
cropped sentences (%)	15	25	141	151	85	99	3
# of sentences in the	135	1302	1104	1175	522	520	188
longest text	62k	786k	283k	311k	1546k	1464k	77k
# of words in the longest	4	6	20	20	10	10	
sentence	783k	920k	1831k	1832k	1176k	1177k	304k
Vocab size	1835k	2107k	3930k	3933k	2619k	2622k	606k
Td	653k	723k	1176k	1177k	848k	850k	250k
# of trainable parameters in	1508k	1649k	2554k	2557k	1899k	1902k	790k
SLCNN small	10	51	150	170	510	440	8
# of trainable parameters in							
SLCNN large							
# of trainable parameters in							
SLCNN+V small							
# of trainable parameters in							
SLCNN+V large							
Training time for a single							
epoch (s)							

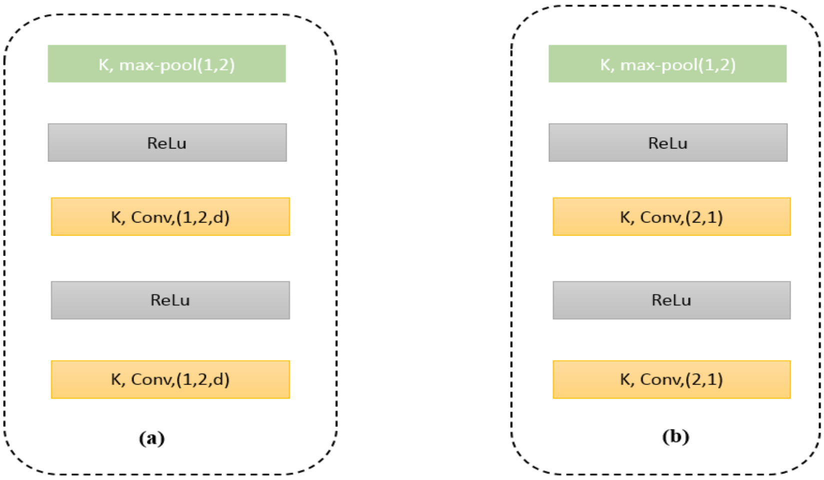

Fig. 5. Convolutional blocks. k is the number of filters. (a) HCB and (b) VCB.

Twitter is often known as a public platform and different Application Programming Interfaces (APIs) have been introduced for its data collection. In this work, we used GetOldTweets[1] and Tweepy[2] to collect the data. Some tools provide access to older tweets. GetOldTweets is a completely free tool for collecting Twitter data that also supports combined search and one-word search features, allowing the access to long-term tweets. This API provides very useful information such as id (str), permalink (str), username (str), to (str), text (str), date (datetime) in Coordinated Universal Time (UTC), retweets (int), favorites (int), mentions (provides str), hashtags (str), and geo (str). The features extracted by GetOldTweets are useful but very few. Therefore, we used Tweepy to extract some other useful features such as the number of followers and followers per person. This powerful tool is also used to collect Twitter data, which uses the OAuth mechanism for authentication

[1] https://pypi.org/project/GetOldTweets3/
[2] https://www.tweepy.org/

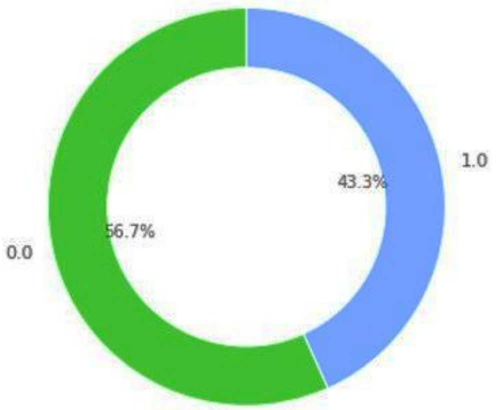

Fig. 6. Ratio of the number of fake news (1) and real news (0) in the collected dataset.

Attempts were made to use keywords such as #Covid - 19, #Corona, and #Corona - virus to gather the information. We also tried to remove some tweets related to inattentive people (people who are less important by other users, which can be measured by the number of followers and retweets) on Twitter. Also, labeling each tweet was done manually, which was a very time-consuming process. The ratio of the number of fake and real news is given in *Fig. 6*. For fake, 16088 news, and for real, 21101 news were collected.

4.3 | Benchmark Datasets

In addition to the Covid-19 dataset, we utilized six datasets covering different classification tasks, compiled by [15]. General specifications are presented in *Table 2*. All data are evenly distributed across class labels. AG and DBPedia are news and ontology classification datasets, respectively. Yelp and Amazon are sentiment classification datasets, where '.P' (polarity) in the dataset names, indicates that the labels are binary, while '.F' (full) means that the labels refer to the number of stars.

Some of the statistical information extracted from the datasets, after the preprocessing step, is summarized in *Table 3*. As presented in the table, by considering $T_s = 46$, the proportion of cropped sentences is between 2 and 2.9%, which shows that the length of sentences in different datasets is almost similar. By contrast, the number of sentences of the documents in the different datasets is quite different. By utilizing *Eq. (1)*, T_d for AG News, DBPedia, Amazon, and Yelp are equal to 4, 6, 10, and 20, respectively. Also, the proportion of cropped documents using relevant T_d is 0.4, 3, 3.6, and 6% for AG News, Amazon, DBPedia, and Yelp, respectively, which means that the variance of the number of sentences in the documents of Yelp is greater than others.

4.4 | Evaluation Protocols

The four metrics of accuracy (ACC), precision, recall, and F1 were used as evaluation criteria for experimental results, which are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{8}$$

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{9}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{10}$$

In these equations, True Positive (TP) is the number of positive instances that are classified as positive; True Negative (TN) is the number of negative instances that are classified as negative; False Positive (FP) is the number of negative instances that are classified as positive; False Negative (FN) is the number of positive instances that are classified as negative.

4.5 | Results

We compared our models with several popular base models, e.g. linear models [15], RNN-based model, i.e., D-LSTM [20], and CNN-based models including classical word-level CNN [23], character-level CNN [15], very deep CNN [24] and CNN with the fastest embedding [73]. Since we aimed to provide new baseline models without using other mechanisms such as attention, such, models were excluded from the comparison. The results are listed in Table 4 based on accuracy. Overall, it can be seen that the proposed models outperformed all the models in half of the datasets, DBPedia, Yelp. P, Covid-19, and Yelp. F. The improvement was especially significant in Yelp datasets. In terms of Amazon datasets, the SLCNN+V was ranked third after VDCNN and character-level CNN with around 94 and 58.1% in Amazon. P and Amazon. F, respectively.

Table 4. Test accuracy (%) of all the models on the datasets. Results marked with * are reported in [11] and others are reprinted from the references.

Models		AG news	DBpedia	Yelp. P	Yelp. F	Amazon. P	Amazon. F	Covid -19
Linear	Bag of words [15]	88.81	96.61	92.24	57.99	90.40	54.64	-
	n-grams [15]	92.04	98.63	95.64	56.26	92.02	54.27	-
	n-grams TF-IDF [15]	92.36	98.69	95.44	54.80	91.54	52.44	-
CNN	Char-level CNN small [15]	84.35	98.02	93.47	59.16	94.50	59.47	-
	Char-level CNN large [15]	87.18	98.27	94.11	60.38	94.49	58.69	-
	VDCNN-29 layers [24]	91.27	98.71	95.72	64.26	95.69	63.00	-
	Word-level CNN [23]*	91.60	98.60	93.50	61.00	-	57.40	-
	FastText [64]	91.50	98.10	93.80	60.40	91.20	55.80	-
RNN	D-LSTM [20]	92.10	98.70	92.60	59.60	-	-	-
Ours	SLCNN small	91.22	98.75	96.03	64.67	93.87	58.03	95.93
	SLCNN large	91.26	98.76	96.01	64.56	93.93	58.02	97.01
	SLCNN+V small	91.45	98.73	96.09	64.46	93.91	58.11	97.33
	SLCNN+V large	91.39	98.76	96.07	64.39	93.94	58.05	96.70

If we look at AG News, n-grams and discriminative-LSTM have achieved better results despite competitive results with other CNN models. One of the main reasons we can mention is the number of sentences in the documents. So that the proposed models perform better in documents with a large number of sentences, i.e., Yelp. Another reason that hinders better performance in Amazon datasets is the very high vocabulary size (see *Table 3*), since we used the word embedding with just over 1M vocabularies in our experiments. The overall accuracy obtained for the proposed models was 97.33%, precision: 95.84%, recall: 93.97%, and f-measure: 94.80%. A higher number of documents were misclassified as real, 450 out of 11300 documents. *Table 5* shows the results obtained for Covid-19 dataset.

Table 5. Test classification report.

	Fake	**Real**
Fake	10850	450
Real	123	9107

5 | Conclusion and Future Works

This paper offers new baseline models for text classification using a SLCNN. The key idea is representing the documents as a 3D tensor to enable the models for sentence-level analysis. The proposed models were compared with the state-of-the-art models such as BOW n-grams, n-grams TF-IDF, character-level CNN small, character-level CNN large, VDCNN-29 layers, word-level CNN, and fastText using several datasets. The results showed that the proposed models have better performance, particularly in the longer documents. The key idea of this approach is to use the spatial information of each sentence in the documents. It is worth noting that considering sentences together can create additional information that is not available in BOW models. In future works, the attention mechanism will be utilized in the proposed models to improve the overall performance. Also, we will work on sentence standardization. We believe that applying a standard form of sentences enables the proposed models to use compositional methods with different 3D filters, due to the 3D structure of the input tensor. Manual and statistical studies of 1000 misclassified documents revealed that the incorrect detection of the negation scope mostly leads to the incorrect classification of these documents. Therefore, as a part of the future work, our next goal is to apply negation scope as a manual feature along with the features selected automatically by the SLCNN approaches. In [19] negative words with 90 patterns were used to detect the negation scope which led to an increase in the performance of the proposed system. Similar patterns can be used to improve our proposed approaches.

References

[1] Zhou, X., & Zafarani, R. (2018). *Fake news: a survey of research, detection methods, and opportunities*. Available at arXiv:1812.00315

[2] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: a data mining perspective. *ACM SIGKDD explorations newsletter*, *19*(1), 22-36. https://doi.org/10.1145/3137597.3137600

[3] Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combating fake news: a survey on identification and mitigation techniques. *ACM transactions on intelligent systems and technology (TIST)*, *10*(3), 1-42.

[4] Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). *Some like it hoax: automated fake news detection in social networks*. Available at arXiv:1704.07506

[5] Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017, September). Truth of varying shades: analyzing language in fake news and political fact-checking. *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2931-2937). Association for Computational Linguistics. https://aclanthology.org/D17-1317

[6] Agag, G. M., & El-Masry, A. A. (2017). Why do consumers trust online travel websites? drivers and outcomes of consumer trust toward online travel websites. *Journal of travel research*, *56*(3), 347-369.

[7] Pang, B., Lee, L., & Vaithyanathan, S. (2002). *Thumbs up? sentiment classification using machine learning techniques*. Available at cs/0205070

[8] Jindal, N., & Liu, B. (2007, May). Review spam detection. *Proceedings of the 16th international conference on world wide web* (pp. 1189-1190). Association for Computing Machinery, New York, NY, United States. https://doi.org/10.1145/1242572.1242759

[9] Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84. https://doi.org/10.1145/2133806.2133826

[10] Russel, S., & Norvig, P. (2013). *Artificial intelligence: a modern approach*. Pearson.

[11] Wang, S. I., & Manning, C. D. (2012, July). Baselines and bigrams: simple, good sentiment and topic classification. *Proceedings of the 50th annual meeting of the association for computational linguistics (Volume 2: Short Papers)* (pp. 90-94). Association for Computational Linguistics.

[12] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551. DOI: 10.1162/neco.1989.1.4.541

[13] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A critical review of recurrent neural networks for sequence learning*. Available at arXiv:1506.00019

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735

[15] Zhang, X., Zhao, J., & LeCun, Y. (2015). *Character-level convolutional networks for text classification*. https://proceedings.neurips.cc/paper/2015/hash /250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html

[16] Feng, G., Li, S., Sun, T., & Zhang, B. (2018). A probabilistic model derived term weighting scheme for text classification. *Pattern recognition letters*, *110*, 23-29. https://doi.org/10.1016/j.patrec.2018.03.003

[17] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality*. Available at arXiv:1310.4546

[18] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). Association for Computational Linguistics.

[19] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. Availabla at arXiv:1406.1078

[20] Yogatama, D., Dyer, C., Ling, W., & Blunsom, P. (2017). *Generative and discriminative text classification with recurrent neural networks*. Availabla at arXiv:1703.01898

[21] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. DOI: 10.1109/5.726791

[22] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, *12*(ARTICLE), 2493-2537.

[23] Kim, Y. (2019). Convolutional neural networks for sentence classification. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1746–1751). Association for Computational Linguistics.

[24] Conneau, A., Schwenk, H., Barrault, L., & Lecun, Y. (2016). *Very deep convolutional networks for text classification*. Availabla at arXiv:1606.01781

[25] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). IEEE.

[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). *Attention is all you need*. Available at arXiv:1706.03762

[27] Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). *A structured self-attentive sentence embedding*. Available at arXiv:1703.03130

[28] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016, June). Hierarchical attention networks for document classification. *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 1480-1489). Association for Computational Linguistics.

Mottaghi et al.| J. Appl. Res. Ind. Eng. 8(Spec. Issue) (2021) 1-17

[29] Wang, S., Huang, M., & Deng, Z. (2018, July). Densely connected CNN with multi-scale feature attention for text classification. *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 4468-4474). AAAI Press.

[30] Castillo, C., Mendoza, M., & Poblete, B. (2011, March). Information credibility on twitter. *Proceedings of the 20th international conference on World wide web* (pp. 675-684). Association for Computing, Machinery, New York, NY, United States. https://doi.org/10.1145/1963405.1963500

[31] Zhang, H., Fan, Z., Zheng, J., & Liu, Q. (2012). An improving deception detection method in computer-mediated communication. *Journal of networks*, *7*(11), 1811-1907.

[32] Zhou, L., Twitchell, D. P., Qin, T., Burgoon, J. K., & Nunamaker, J. F. (2003, January). An exploratory study into deception detection in text-based computer-mediated communication. *Proceedings of the 36th annual Hawaii international conference on system sciences*. IEEE. DOI: 10.1109/HICSS.2003.1173793

[33] Chang, C., Zhang, Y., Szabo, C., & Sheng, Q. Z. (2016, December). Extreme user and political rumor detection on twitter. *International conference on advanced data mining and applications* (pp. 751-763). Springer, Cham. https://doi.org/10.1007/978-3-319-49586-6_54

[34] Aker, A., Derczynski, L., & Bontcheva, K. (2017). *Simple open stance classification for rumour analysis*. Available at arXiv:1708.05286

[35] Ruchansky, N., Seo, S., & Liu, Y. (2017, November). Csi: a hybrid deep model for fake news detection. *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 797-806). Association for Computing, Machinery, New York, NY, United States. https://doi.org/10.1145/3132847.3132877

[36] Giasemidis, G., Singleton, C., Agrafiotis, I., Nurse, J. R., Pilgrim, A., Willis, C., & Greetham, D. V. (2016, November). Determining the veracity of rumours on Twitter. *International conference on social informatics* (pp. 185-205). Springer, Cham. https://doi.org/10.1007/978-3-319-47880-7_12

[37] Vosoughi, S. (2015). *Automatic detection and verification of rumors on Twitter* (Doctoral dissertation, Massachusetts Institute of Technology). Retrieved from https://lsm.media.mit.edu/papers/Soroush_Vosoughi_PHD_thesis.pdf

[38] Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, *32*(2), 604-624. DOI: 10.1109/TNNLS.2020.2979670

[39] Zhang, Y., Meng, J. E., Venkatesan, R., Wang, N., & Pratama, M. (2016, July). Sentiment classification using comprehensive attention recurrent models. *2016 international joint conference on neural networks (IJCNN)* (pp. 1562-1569). IEEE. DOI: 10.1109/IJCNN.2016.7727384

[40] Rojas-Barahona, L. M. (2016). Deep learning for sentiment analysis. *Language and linguistics compass*, *10*(12), 701-719. https://doi.org/10.1111/lnc3.12228

[41] Deng, L., & Liu, Y. (Eds.). (2017). *Deep learning in natural language processing*. Springer, Singapore.

[42] LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision* (pp. 319-345). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-46805-6_19

[43] Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011, June). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. *CVPR 2011* (pp. 3361-3368). IEEE. DOI: 10.1109/CVPR.2011.5995496

[44] Tompson, J. J., Jain, A., LeCun, Y., & Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. *Proceedings of the 27th international conference on neural information processing systems* (pp. 1799-1807). MIT Press.

[45] Chen, H., Xie, L., Leung, C. C., Lu, X., Ma, B., & Li, H. (2016). Modeling latent topics and temporal distance for story segmentation of broadcast news. *IEEE/ACM transactions on audio, speech, and language processing*, *25*(1), 112-123.

[46] Zeiler, M. D., & Fergus, R. (2013). *Stochastic pooling for regularization of deep convolutional neural networks*. Available at arXiv:1301.3557

[47] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), 1904-1916. DOI: 10.1109/TPAMI.2015.2389824

[48] Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., ... & Tang, X. (2014). *Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection*. Available at arXiv:1409.3505

[49] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: a review. *Neurocomputing*, *187*, 27-48. https://doi.org/10.1016/j.neucom.2015.09.116

[50] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., & Yu, P. S. (2018). *TI-CNN: convolutional neural networks for fake news detection*. Available at arXiv:1806.00749

[51] Ajao, O., Bhowmik, D., & Zargari, S. (2018, July). Fake news identification on twitter with hybrid cnn and rnn models. *Proceedings of the 9th international conference on social media and society* (pp. 226-230). Association for Computing Machinery, New York, NY, United States. https://doi.org/10.1145/3217804.3217917

[52] Dragoni, M., & Petrucci, G. (2017). A neural word embeddings approach for multi-domain sentiment analysis. *IEEE transactions on affective computing*, *8*(4), 457-470. DOI: 10.1109/TAFFC.2017.2717879

[53] Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. Available at arXiv:1409.0473

[54] Long, Y. (2017). Fake news detection through multi-perspective speaker profiles. *The eighth international joint conference on natural language processing* (Volume 2: Short Papers). Asian Federation of Natural Language Processing, Taipei, Taiwan.

[55] Karimi, H., & Tang, J. (2019). *Learning hierarchical discourse-level structure for fake news detection*. Available at arXiv:1903.07389

[56] Chauhan, A., Babu, M., Kandru, N., & Lokegaonkar, S. (2018). *Empirical Study on convergence of capsule networks with various hyperparameters*. Retrieved from https://people.cs.vt.edu/~bhuang/courses/opt18/projects/capsule.pdf

[57] Sabour, S., Frosst, N., & Hinton, G. E. (2017). *Dynamic routing between capsules*. https://proceedings.neurips.cc/paper/2017/hash/2cad8fa47bbef282badbb8de5374b894-Abstract.html

[58] Hinton, G. E., Sabour, S., & Frosst, N. (2018, February). Matrix capsules with EM routing. Paper presented at the metting of International conference on learning representations, Vancouver Convention Center, Vancouver CANADA.

[59] Deng, F., Pu, S., Chen, X., Shi, Y., Yuan, T., & Pu, S. (2018). Hyperspectral image classification with capsule network using limited training samples. *Sensors*, *18*(9), 3153. https://doi.org/10.3390/s18093153

[60] Iesmantas, T., & Alzbutas, R. (2018, June). Convolutional capsule network for classification of breast cancer histology images. *International conference image analysis and recognition* (pp. 853-860). Springer, Cham. https://doi.org/10.1007/978-3-319-93000-8_97

[61] De La Escalera, A., Moreno, L. E., Salichs, M. A., & Armingol, J. M. (1997). Road traffic sign detection and classification. *IEEE transactions on industrial electronics*, *44*(6), 848-859. DOI: 10.1109/41.649946

[62] Paoletti, M. E., Haut, J. M., Fernandez-Beltran, R., Plaza, J., Plaza, A., Li, J., & Pla, F. (2018). Capsule networks for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, *57*(4), 2145-2160. DOI: 10.1109/TGRS.2018.2871782

[63] Goli, A., Zare, H. K., Tavakkoli-Moghaddam, R., & Sadeghieh, A. (2019). Application of robust optimization for a product portfolio problem using an invasive weed optimization algorithm. *Numerical algebra, control & optimization*, *9*(2), 187-209. DOI: 10.3934/naco.2019014

[64] Goli, A., Tirkolaee, E. B., & Aydın, N. S. (2021). Fuzzy integrated cell formation and production scheduling considering automated guided vehicles and human factors. *IEEE transactions on fuzzy systems*, *29*(12), 3686-3695. DOI: 10.1109/TFUZZ.2021.3053838

[65] Goli, A., & Malmir, B. (2020). A covering tour approach for disaster relief locating and routing with fuzzy demand. *International journal of intelligent transportation systems research*, *18*(1), 140-152. https://doi.org/10.1007/s13177-019-00185-2

[66] Goli, A., Zare, H. K., Moghaddam, R., & Sadeghieh, A. (2018). A comprehensive model of demand prediction based on hybrid artificial intelligence and metaheuristic algorithms: a case study in dairy industry. *Journal of industrial and systems engineering*, 11, 190-203.

[67] Goli, A., Khademi-Zare, H., Tavakkoli-Moghaddam, R., Sadeghieh, A., Sasanian, M., & Malekalipour Kordestanizadeh, R. (2021). An integrated approach based on artificial intelligence and novel meta-heuristic algorithms to predict demand for dairy products: a case study. *Network: computation in neural systems*, *32*(1), 1-35. https://doi.org/10.1080/0954898X.2020.1849841

[68] Lotfi, R., Mardani, N., & Weber, G. W. (2021). Robust bi-level programming for renewable energy location. *International journal of energy research*, *45*(5), 7521-7534. https://doi.org/10.1002/er.6332

[69] Lotfi, R., Yadegari, Z., Hosseini, S. H., Khameneh, A. H., Tirkolaee, E. B., & Weber, G. W. (2020). A robust time-cost-quality-energy-environment trade-off with resource-constrained in project management: a case study for a bridge construction project. *Journal of industrial & management optimization*. *13*(5), 1-22. DOI: 10.3934/jimo.2020158

[70] Lotfi, R., Nayeri, M., Sajadifar, S., & Mardani, N. (2017). Determination of start times and ordering plans for two-period projects with interdependent demand in project-oriented organizations: a case study on molding industry. *Journal of project management*, *2*(4), 119-142. DOI: 10.5267/j.jpm.2017.9.001

[71] Lotfi, R., Mehrjerdi, Y. Z., Pishvaee, M. S., Sadeghieh, A., & Weber, G. W. (2021). A robust optimization model for sustainable and resilient closed-loop supply chain network design considering conditional value at risk. *Numerical algebra, control & optimization*, *11*(2), 221-253.

[72] Kingma, D. P., & Ba, J. (2014). *Adam: a method for stochastic optimization*. Available at arXiv:1412.6980

[73] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). *Bag of tricks for efficient text classification*. Available at arXiv:1607.01759