IJICTR
International Journal of Information & Communication Technology Research

Volume 16- Number 4 – 2024 (33 -43)

Persian Rumor Detection Using a Multi-Classifier Fusion Approach

Alireza Mansouri*
Information Technology Faculty
ICT Research Institute
Tehran, Iran
amansuri@itrc.ac.ir

Maryam Mahmoudi
Information Technology Faculty
ICT Research Institute
Tehran, Iran
mahmoudy@itrc.ac.ir

Mojgan Farhoodi
Information Technology Faculty
ICT Research Institute
Tehran, Iran
farhoodi@itrc.ac.ir

Mohammadreza Mirsarraf
Information Technology Faculty
ICT Research Institute
Tehran, Iran
mirsaraf@itrc.ac.ir

Received: 27 January 2024 – Revised: 21 April 2024 - Accepted: 18 May 2024

Abstract— **During the last few years, rumor and its rapid diffusion via social media have affected public opinions, even in some important such as presidential elections. One of the main approaches for rumor detection methods is based on content and natural language processing. Despite considerable improvement made in this regard in the English language, unfortunately, we have not witnessed enough progress in the Persian language, mainly due to a lack of datasets in this area. The main novelty of this paper is combining different learning methods to consider the classification problem from different aspects and combine the classifiers' results to achieve a reasonable final result. In the proposed method, each classifier is assigned a weight depending on its f-measure value; thus, the final fused result is closer to the performance of the best classifier. When news samples have various characteristics, and the best classifier is not pre-determined, this fusion method is more beneficial. Therefore, as the conclusion of this research, compared to a single rumor detection method, the fusion of classifiers could be used to achieve better results when the news samples have various characteristics.**

Keywords: Rumor detection, Machine learning, Content-based text classification, Deep learning, Multi-classification

Article type: Research Article

Publisher: ICT Research Institute

* Corresponding author

[DOI: 10.61186/itrc.16.4.33]

I. Introduction

Social media have many advantages and disadvantages. The advantages include low cost, easy access, and rapid information propagation; however, one of the main disadvantages is the possibility of widespread fake news and rumors [1]. Fake news disseminations in online media and social networks have caused erosions to democracy, justice, and public trust; therefore, we face increased demand for fake news and/or rumor detection and intervention. [2].

When a social media user receives a rumor, his opinion may be affected depending on various factors. These factors include the user's trust in the sender, the number of times he has received it, the social network structure that affects links and segregations [3, 4], and psychological parameters discussed as the opinion formation models, such as the social impact model of opinion formation [5] and Deffuant model of opinion formation [6]. The emotional aspects of the news also may affect the users' opinions [7]. A viral rumor on social networks may change the opinion of the majority of the society, or in terms of the physicians, it may cause a phase transition [8, 9]. Therefore, a planned opinion phase transition on social media may manipulate and alter public opinion. A recently published study [10] shows that 28 countries have had organized social media manipulation campaigns in 2017, which increased to 48 countries in 2018, and to 70 countries in 2019, mainly using Facebook and Twitter. Among the public opinion manipulations are the 2020 US presidential election [11], the 2019 Portuguese election [12], the 2017 French presidential election [13], the 2016 US presidential election [14-16], and the 2016 UK European Union membership referendum (Brexit) [17].

Detecting rumors and controlling their impacts on society is essential to achieve a more trustable and purified cyber society. Due to the large number of posts exchanged on online social media, using automatic methods to (even roughly) detection of rumors is inevitable. Recently, natural language processing (NLP), machine learning, and social network analysis have been widely used for rumor detection [1, 2, 18].

In this research, we used three classification methods based on 1) lingual-based features, 2) word frequency-based features (Bag of Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF)), and 3) word embedding-based features. The results of these classifiers are sometimes different, and some samples are misclassified by some classifiers due to a variety of reasons, such as the use of different features and different parameters to adjust the algorithm. Therefore, as the main contribution of in this study, we used Multi-classification and fusing the classifiers' results to enhance the overall performance of the proposed multi-classification method.

The rest of this paper is structured as follows: Section II introduces some related works, Section III explains the proposed method and related basic concepts, Section IV reports the results of the experiments, Section V is dedicated to the discussion, and finally Section VI concludes the paper.

II. Related Works

Recently, several studies have been conducted on rumor detection in English and some other languages using various approaches. Some studies have used various lingual features of the news text for rumor detection, as summarized in Table I. However, unfortunately, few studies have been conducted on Persian rumor detection, as among the studies listed in Table I, just one study has focused on Persian news [19], and others have studied English news. Our research focuses on Persian news using the lingual feature-based approach; therefore, our research is also shown in the table, specifying which lingual features we have used.

One of the problems in studies in this field on Persian language is the lack of sufficient datasets. In [20], to solve the lack of Persian datasets problem, a machine translator has been used to translate English tweets into Persian. However, this approach imposes an error on the system. In [21], a Persian rumor detection on Twitter based on context-based features has been reported, and [19] analyzed the content of the original rumor and introduced informative content features to early identifying Persian rumors on Twitter and Telegram, i.e., when it is published on news media but has not yet spread on social media.

No study is reported for Persian rumor detection using N-gram and TF-IDF; however, some studies are reported on other languages. In [31], N-gram and TF-IDF have been used for rumor detection. Wynne et al. [32] have used N-gram and TF-IDF for rumor detection. Oriola has also used some content-based methods for rumor detection, including N-gram and TF-IDF; we are interested in this study.

Few studies have also been conducted on Persian rumor detection using deep learning. Samadi et al. [33] have proposed two different architectures for rumor detection using the BERT pre-trained model. Also Samadi et al. in 2023 [34] used content-based features and semantic textual features within a deep CNN framework. Jahanbakhsh-Nagadeh et al. [35, 36] have used semantic features and ParsBERT [37], a monolingual BERT for the Persian language, for Persian rumor verification. Mottaghi et al. [38] have used a convolutional neural network (CNN) model for their deep learning approach to detect Persian rumors. Ghayoome et al. [39] have developed a deep cross-lingual contextualized language model for fake news detection. Sadr et al. [40] have used a hybrid of Long Short-Term Memory (LSTM) and bidirectional LSTM (BLSTM) for Persian fake news detection.

Ensembling classifiers have been used in various applications [41]. However, few studies have focused on ensembling rumor classifiers [42].

III. Methods

This section describes the methods we used in this research.

A. Datasets

In this study, two data sets were used for evaluating rumor detection classification methods, including Sepehr_RumTel01 [43, 44] and KNTUPT [45].

[DOI: 10.61186/itrc.16.4.33]

TABLE I. LINGUAL-BASED FEATURES AND RELATED WORKS

Attribute Type	Feature	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[19]	Our Study
Quantity	Number of characters			✓						✓	✓	✓
	Number of words	✓	✓	✓	✓	✓				✓	✓	✓
	Number of noun phrases	✓										
	Number of sentences	✓	✓	✓	✓					✓	✓	✓
	Number of paragraphs							✓		✓		
Complexity	Average number of characters per word	✓	✓	✓	✓					✓		✓
	Average number of words per sentence	✓	✓	✓	✓	✓				✓		✓
	Average number of clauses per sentence	✓			✓							
	Average number of punctuations per sentence	✓	✓	✓	✓						✓	✓
	Average number of Name Entity per sentence										✓	
Uncertainty	#/% Modal verbs (e.g., “shall”)	✓	✓	✓	✓							✓
	#/% Certainty terms (e.g., “never” and “always”)	✓	✓	✓	✓		✓			✓	✓	✓
	#/% Generalizing terms (e.g., “generally” and “all”)		✓									✓
	#/% Tentative terms (e.g., “probably”)		✓	✓			✓			✓	✓	✓
	#/% Numbers and quantifiers			✓							✓	✓
	#/% Question marks			✓				✓		✓	✓	✓
	#/% Inferential words/phrase (e.g., “as a result”)										✓	✓
Subjectivity	#/% Biased lexicons (e.g., “attack”)									✓		
	#/% Subjective verbs (e.g., “feel” and “believe”)	✓				✓					✓	✓
	#/% Report verbs (e.g., “announce”)									✓		✓
	#/% Factive verbs (e.g., “observe”)									✓		
	#/% Motion verbs (e.g., “fall”, “shake”)										✓	
Non-immediacy	#/% Passive voice	✓	✓		✓							
	#/% Self reference: 1st person singular pronouns	✓	✓	✓	✓	✓						✓
	#/% Group reference: 1st person plural pronouns	✓	✓	✓	✓	✓					✓	✓
	#/% Other reference: 2nd and 3rd person pronouns	✓	✓	✓		✓					✓	✓
	#/% Quotations			✓								
Sentiment	#/% Positive words	✓	✓	✓	✓	✓	✓			✓	✓	✓
	#/% Negative words	✓	✓	✓	✓	✓	✓			✓	✓	✓
	#/% Anxiety/angry/sadness words						✓			✓	✓	✓
	#/% Exclamation marks			✓						✓		
	Content sentiment polarity									✓	✓	
	Emotiveness (The ratio of the sum of adjectives, adverbs and sensory/motion verbs to total words)										✓	
	Newsworthy (text-enhancing components)										✓	
Diversity	Lexical diversity: #/% unique words or terms	✓	✓	✓	✓	✓				✓		
	Content word diversity: #/% unique content words	✓	✓			✓				✓		
	Redundancy: #/% unique function words	✓	✓	✓		✓				✓		
	#/% Unique nouns/verbs/adjectives/adverbs									✓	✓*	✓*
Informality	#/% Typos (misspelled words)	✓			✓	✓					✓	
	#/% Swear words/netspeak/assent/non fluencies/fillers									✓	✓	✓
	Start/End phrase (e.g., “Urgent!”, “Please Share!”)										✓	✓
	Consecutive letter/Word (e.g., “very very important!”)										✓	✓
	Emoji										✓	
Specificity	Temporal/spatial ratio	✓	✓				✓				✓	
	Sensory ratio	✓	✓		✓		✓			✓		
	Causation terms		✓				✓			✓		
	Exclusive terms		✓									
Readability (e.g., Flesch-Kincaid and Gunning-Fog index)				✓				✓	✓	✓	✓	

*: The average of nouns/ verbs/ adjectives/ adverbs

1) Sepehr_RumTel01 Dataset

The Sepehr_RumTel01 dataset[1] [43, 44] is taken from the telegram channels of three Iranian websites, Gomaneh (Gomaneh.com), WikiHoax (wikihoax.org), and Anti-Rumor (Shayeaat.ir). This dataset contains 1911 news, comprising 680 rumors and 1231 truthful news. This dataset is a simple Excel file containing a news “text” column and a binary (0 for rumor or 1 for truthful) “label” column.

2) KNTUPT Dataset

The KNTUPT dataset[2] [45] is collected from Twitter. It includes 3593704 tweets that were published in the period from November 24 to December 8, 2017. The tweets are based on 60 news about the Kermanshah earthquake, in the west of Iran, with a magnitude of 6.3. The new set is mentioned on the Shayeaat website, a fact-checking website. In this dataset, 4343 out of 3593703 news (about one percent) are rumors. Since it was very unbalanced, we extracted all the rumors and randomly selected 2*4343=8686 real news; therefore,

[1] The dataset is available on https://data.mendeley.com/datasets/jw3zwf8rdp/3.

[2] The dataset is available on https://trlab.ir/res.php?resource id=3.

[DOI: 10.61186/itrc.16.4.33]

the total size of this dataset was reduced to 13029 for our experience, one-third for the rumors, and two-thirds for the real news. The dataset is presented in a Microsoft SQL Server format containing 22 features in three classes: content features, demographic features, and structural features. We just used the texts and rumor binary labels from this dataset and extracted the required features.

B. Classifiers Preparation

This research recruits various classifiers and their fusion. Therefore, we first prepared each classifier according to the general two-phased process shown in Fig. 1. As the figure shows, in phase 1 (learning phase), the classifier model is trained with the labeled dataset. Labels indicate whether each post is a rumor or not. The learning process depends on the classifier model. After the learning phase, the classifier is ready to predict or label the input samples.

C. The Multi-classifier Architecture

Our experiment is conducted based on a simple multi-classifier architecture shown in Fig. 2. This figure shows the components of our proposed multi-classifier architecture composed of "data cleaning/ preprocessing", the classifiers, "fuser", and "evaluation" components. The figure also shows the data flow between the components. The following sections explain the components of this architecture.

D. Data cleaning/ preprocessing

The "data cleaning/ preprocessing" component prepares the input dataset to be processed in the next steps. This component normalizes the posts by removing unnecessary characters, unifying some Persian characters with more than one form, and unifying the various forms of typewriting, which is one of the Persian typewriting challenges.

Furthermore, this component extracts some information such as word stems, segmentation, and sentence splitting. For using the classification methods based on N-gram, TF-IDF, and LSTM, stop words and punctuation marks are also removed from the input dataset, whereas the stop words remain for lingual feature based classifier because some of the features are extracted from the stop words and punctuation marks. This component uses Hazm , a free Python library for Persian language processing based on NLTK library. The learned classifiers receive preprocessed posts from pre-processed dataset without labels and label each post to send to the "fuser" component.

E. Lingual features based classification

The prevailing way of characterizing and detecting rumors and fake news based on lingual features relies on the lingual features in various language levels: lexicon, syntax, discourse, and semantics [2]. In this technique, the following feature groups are mostly used:

- Linguistic features: related to the characters, words, and sentences of the post, as well as part of speech (POS) of the words and phrases.
- Psycho-linguistic features: dealing with the sentimental analysis of the post, extracting positive, negative, and neutral sentiments of the sentences of the post.
- Stylometric features: related to writing style, including the percentage of the numbers and the particles of whole the post, punctuation marks, long words, short words, and some other similar features.

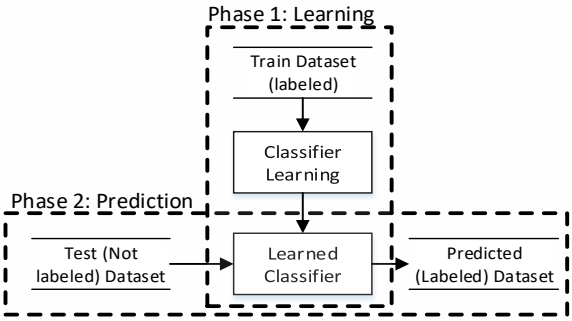


Figure 1. General two phased learning prediction of classifiers

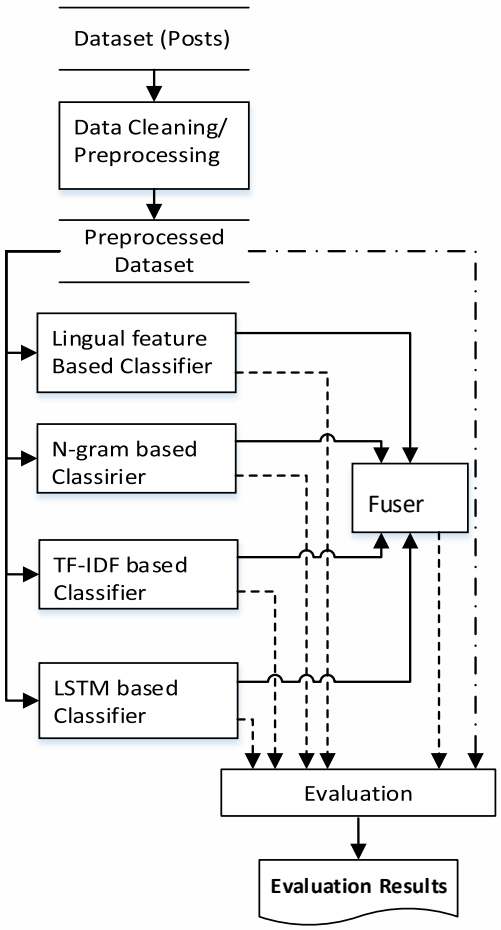


Figure 2. Architecture of the experiment

This component extracts the features using text processing modules and normalizes the extracted statistical features to the range [0..1] using the min-max normalization method. The feature set we used in this method is presented in the column "Our Study" of Table I.

In this component, we used different supervised algorithms and compared the results to choose the best one. Table II and Table III show the algorithms we

[DOI: 10.61186/itrc.16.4.33]

considered for this component and the results obtained from each one on both datasets, respectively. Overall, as the tables show, the Random Forest (RF) algorithm performs better than other algorithms; thus, in the rest of the paper, we report just the results of RF algorithm. We used the free scikit-learn library in Python to implement the classification methods.

F. N-gram and TF-IDF based classifiers

N-gram and TF-IDF based classifiers have similar fundamentals to the bag of words models. From a content viewpoint, documents have a similar classification result if they have similar content. In addition, much can be learned from the content alone about the document content. The first step in a bag of word implementation is vocabulary management. The length of the document vector is equal to the number of known words. Each document may contain a small number of known words in the vocabulary. It results in a vector with a high number of zeros called a scattered vector or scattered representation. The scattered vectors require more memory and computational resources when modeling, and a large number of positions or dimensions can make the modeling process very challenging for traditional algorithms. Thus, when using a bag of word model, it is necessary to reduce the size of the words. A simple text cleaning that can be used as the first step includes a) ignoring punctuation, b) ignoring stop words, c) misspelling correction, and d) replacing words to their stems using a stemmer.

After selecting the words, the occurrence of the words in the sample documents should be scored. In this study, two scorings are used:

- word frequency for N-gram based classification
- TF-IDF for TF-IDF based classification.

For N-gram classifications, we examined uni-gram and bi-gram and the results of uni-gram were better than bi-gram. Thus, our implementation of N-gram classification is indeed based on uni-gram.

Fig. 3 shows the general architecture of the bag of words classification we used in this study. Although we used the SVM algorithm as the classification algorithm for our bag of words methods due to its better performance, other classification algorithms could also be used instead.

TABLE II. RESULTS OF VARIOUS MACHINE LEARNING ALGORITHMS ON SEPEHR_RUMTEL01 DATASET

Algorithm	Precision	Recall	F-Measure	Accuracy
SVM	**0.85**	0.69	0.71	0.80
Log-Regression	0.84	0.71	0.73	0.80
Random Forest	0.84	**0.80**	**0.81**	0.83
Naïve Base	0.76	0.69	0.70	0.77
KNN	0.81	0.73	0.75	0.79
Decision Tree	0.75	0.73	0.74	0.77
LDA (Linear Discriminant Analysis)	**0.85**	0.77	0.79	**0.84**

TABLE III. RESULTS OF VARIOUS MACHINE LEARNING ALGORITHMS ON KNTUPT DATASET

Algorithm	Precision	Recall	F-Measure	Accuracy
SVM	0.41	0.50	0.45	0.81
Log-Regression	0.54	0.50	0.45	0.82
Random Forest	**0.93**	**0.85**	**0.88**	**0.94**
Naïve Base	0.60	0.64	0.43	0.44
KNN	0.86	0.82	0.84	0.91
Decision Tree	0.82	0.83	0.83	0.90
LDA (Linear Discriminant Analysis)	0.63	0.51	0.48	0.82

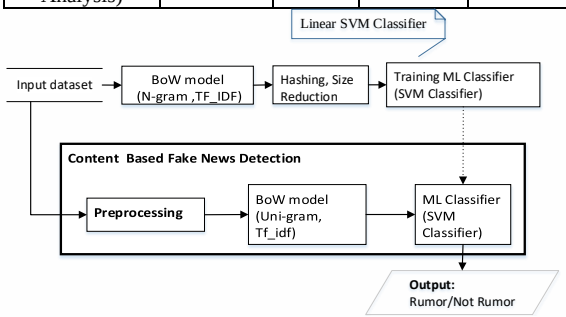


Figure 3. Rumor detection using bag of words algorithm

G. Classification based on deep learning

Deep learning is a subcategory of machine learning, a neural network with more than two layers of hidden units or neurons. The deep networks are deep in terms of the number of neuron layers in the network. Generally, deep learning displays relatively high precision and exactness in rumor detection [21]; however, it uses more memory [22].

LSTM, a widely used deep learning architecture, is a neural network framework based on the recurrent neural network (RNN). LSTM aims to deal with the vanishing gradient problem present in traditional RNNs. Thus, LSTM is a special kind of recurrent neural network capable of learning long term dependencies in data. This is achieved because the recurring module of the model has a combination of four layers interacting with each other. Since back propagation in RNN takes a while, as a progressed variation of RNN, LSTM overcomes this limitation of traditional RNN with its property of remembering "Short Term Memories" for "Long periods." [23]. Compared to the RNN neurons, which have two gates, input and output gates, LSTM neurons have an additional forget gate, in the hidden layers. Forget gate makes LSTM capable of having the property of memorization, as shown in Fig. 4.

A common LSTM unit remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Forget gates decide what information to discard from a previous state by assigning a previous state, compared to a current input, a value between 0 and 1. A (rounded) value of 1 means to keep the information, and a value of 0 means to discard it. Input gates decide which pieces of new information to store in the current state, using the same system as forget gates. Output gates control which pieces of information in the current state to output by assigning a value from 0 to 1 to the

[DOI: 10.61186/itrc.16.4.33]

information, considering the previous and current states. Selectively outputting relevant information from the current state allows the LSTM network to maintain useful, long-term dependencies to make predictions, both in current and future time-steps.

For implementing LSTM, we used Tensorflow and Keras libraries in Python. Due to the variable post sizes of the dataset, we fixed the size to the average of post sizes plus their standard deviation (146 and 25 for Sepehr_RumTel01 and KNTUPT, respectively). The shorter input posts were filled with paddings, and the longer input posts were truncated. Also, we used Word2Vec embedding layer Persian FastText with a dimension of 300 for each input word.

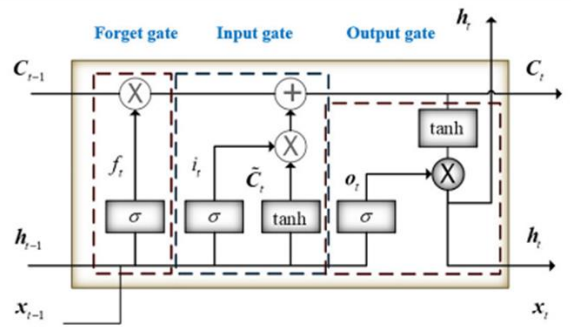


Figure 4. Basic structure of the LSTM model

We used bidirectional LSTM with an output size of 32. To train the model, we divided the dataset to train and test sequences. The size of train sequence was 0.2 of dataset and the size of test sequence was 0.8 of dataset. The batch size of training for LSTM was 256 and the number of epochs was 20. Fig. 5 shows the block diagram of LSTM rumor detector. As the figure shows, the word presentation in the input of LSTM is Fast text word embedding, developed by Facebook on more than 157 languages, including Persian.

BERT could also be used in this component, similar to [46], in which a pertained BERT base decoder for text feature extractor has been used. The padded and tokenized text is passed into the BERT model to receive word vectors of dimension 768 then they have used LSTM as we did on the top of Word2Vec like our method.

H. Fast text embedding algorithm

Word embedding is a way to convert textual information into numeric form, which in turn can be used as input to machine learning algorithms. One major draw-back for word-embedding techniques like Word2Vec was its inability to deal without of corpus words. These embedding techniques treat words as the minimal entities and try to learn their respective embedding vector. Therefore, if a word does not appear in the corpus, Word2Vec fails to get its vectorized representation. However, FastText follows the same skipgram and CBoW (Continuous BoW) model like Word2Vec. FastText is a modified version of Word2Vec, treating each word as composed of N-grams. In Word2Vec each word is represented as a bag of words, but in FastText each word is represented as a bag of N-gram characters. This results in better representation for morphological languages like Persian; thus, we used this embedding as the input of LSTM.

I. Fusion

Supervised classifiers are trained on training datasets and usually are tested on input data with similar patterns to the training data. Moreover, various classifiers have different performances on test data, depending on both the inherent nature of the classifier and the pattern of the input test data. Therefore, the performance of various classifiers may differ on a particular input data. In such a case, as shown in Fig. 6, ensembling the various classifiers to achieve a fused result could enhance the performance [24].

Several fusion methods could be used for ensembling classifiers, mainly depending on the nature of classifiers' output, e.g., probabilistic, ranking, multilabel, or binary. For the case of binary classifiers similar to this research, the foremost fusion method is "weighted majority voting".

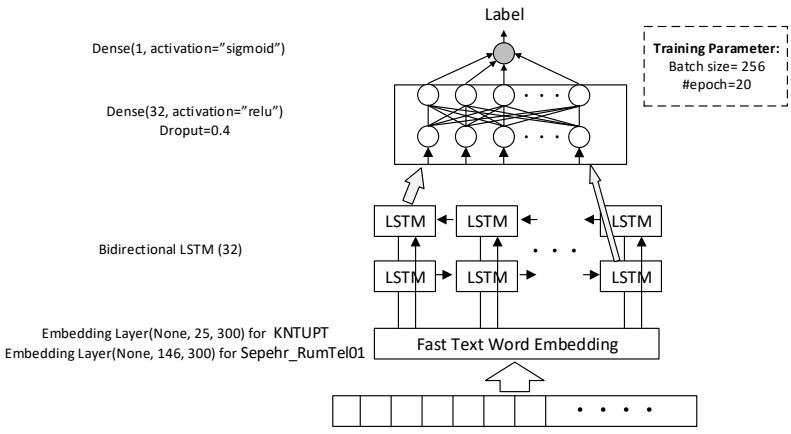
The weighted voting assigns weight w_j to each of N classifiers, C_1 .. C_N, and the final label y is calculated as:

$$y = \underset{i}{\operatorname{argmax}} \sum_{j=1}^{m} w_j X_A(C_j(x) = i),$$

where, A is the set of unique class labels, $1..m$, and X_A is the characteristic function $[C_j(x) = i \in A]$.

Designing a combined classifier requires special care in the choice of individual classifiers to achieve higher classification performance and have more robust algorithms. A multi-classifier usually consists of two parts:

- A set of individual classifiers
- The method of selection or combination for the final classification.



[DOI: 10.61186/itrc.16.4.33]

Figure 5. Block diagram of LSTM

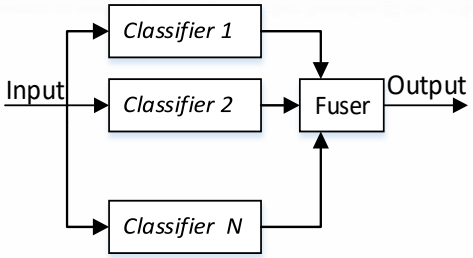


Figure 6. Ensemble classifiers

To achieve a high performance fusion output, in addition to high performance individual classifiers, their diversity is significant. The diversity implies that classifiers consider the subject of classification from various viewpoints or feature sets for classification. In our method, the four rumor classifiers focus on the input posts from two viewpoints. On the one hand, the N-gram, TF-IDF, and LSTM classifiers deal with the words in the posts and consider them as the features for classification; on the other hand, the lingual feature based classifier focuses on the lingual features, the selected features of writing style.

The classifiers in this study are binary classifiers that predict whether a post is a "rumor" or not. The fusion method we used for these binary labeled classifiers is the weighted majority voting, whose weights are F1-score. The F1-score is an overall classification performance metric, a harmonic mean of the precision and recall metrics. More details of the mentioned metrics will be described in Section "IV. Results".

J. Evaluation

The evaluation module of the architecture receives the predictions on every post generated by each trained classifier separately and also from fuser module. Then, it calculates and reports the performance evaluation metrics for each classifier module and the fuser module to compare.

IV. Results

For each classification method mentioned in Section III, we trained a model from labeled data and used it to classify new (unseen) posts whether it is "rumor" or not, using 10-fold stratified cross validation. By 10-fold cross validation, the dataset is partitioned into 10 folds with (roughly) the same number of samples; then, one fold is kept for test, and the other nine folds are used for training the model. After repeating this process for ten times, every sample is chosen once for the test, and the metrics could be measured for all samples. The stratified version of 10-fold cross validation guarantees that train and test folds in each repetition contain (roughly) the same proportion of class labels.

To evaluate the performance classification methods, we used the following commonly used metrics:

- Accuracy: the percentage of correct predictions for the test data, calculated by dividing the number of correct predictions by the number of total predictions.
- Precision: the fraction of relevant examples (true positives) among all of the examples that were predicted to belong in a certain class.
- Recall: the fraction of examples that were predicted to belong to a class with respect to all of the examples that truly belong in the class.
- F-Measure: The adjusted F-Measure allows us to weigh precision or recall more highly if it is more important for our use case. Its formula is slightly different:

$$F_\beta = (1+\beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall}$$

F1 (when β=1) is the harmonic mean of precision and recall we used in this study.

Since we are interested in the classifier's performance in detecting rumors, we report the measured metrics on the "rumor" class. Table IV and Table V show the classification methods' results on Sepehr_RumTel01 and KNTUPT datasets, respectively. The measured values have been rounded to two decimal points. The bar charts of Fig. 7 and Fig. 8 also visualize the same evaluation results of Table IV and Table V, respectively.

TABLE IV. Evaluation Metrics of Rumor Detection Classifiers on Sepehr_RumTel01 Dataset (680 Rumors among 1911 Posts)

Method	Algorithm	Feature	P	R	F1	Acc.
Lingual Features-based	**RF**	**As specified in Table I ("Our Study" column)**	0.83	0.81	0.80	0.84
Word Frequency-based	**SVM**	**BoW (N-gram)**	0.89	0.78	**0.83**	**0.89**
		TF-IDF	0.89	0.51	0.65	0.80
Word Embeding-based	**LSTM**	**Fast-Text**	0.78	**0.85**	0.81	0.86
WMV*			**0.91**	0.77	**0.83**	**0.89**

*: Weighted Majority Voting

TABLE V. Evaluation Metrics of Rumor detection Classifiers on KNTUPT dataset (4343 Romors among 13029 Posts)

Method	Algorithm	Feature	P	R	F1	Acc.
Lingual Features-based	**RF**	**As specified in Table I ("Our Study" column)**	0.93	0.85	0.88	0.94
Word Frequency-based	**SVM**	**BoW (N-gram)**	**0.99**	**0.98**	**0.98**	**0.99**
		TF-IDF	0.94	0.92	0.93	0.96
Word Embeding-based	**LSTM**	**Fast-Text**	0.91	0.93	0.92	0.95
WMV*			0.98	0.96	0.97	0.98

[DOI: 10.61186/itrc.16.4.33]

*: Weighted Majority Voting

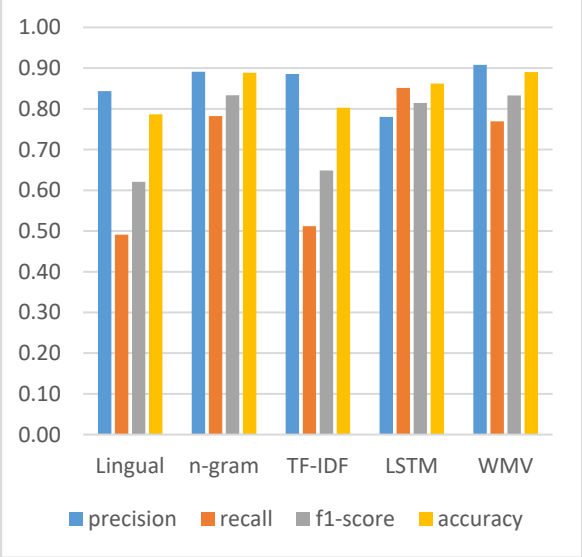


Figure 7. Measured metrics of classifier methods on Sepehr_RumTel01

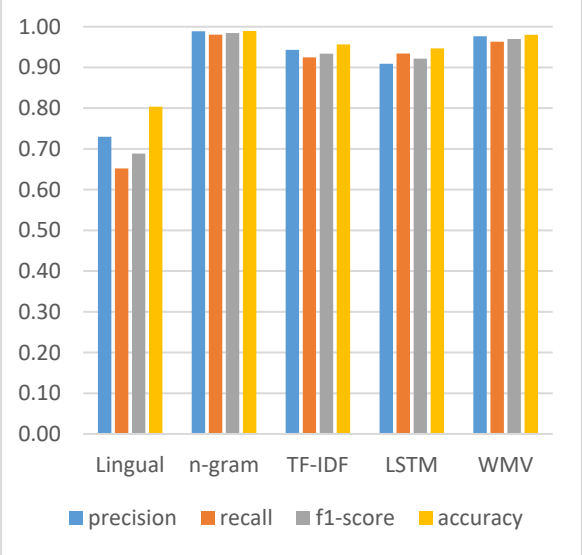


Figure 8. Measured metrics of classifier methods on KNTUPT

V. Comparison and Discussion

Before considering the fusion results, the measured metrics results from methods separately (Table IV, Table V, Fig. 7, and Fig. 8) show that the methods dealing with the words of the news as features perform better than classification based on lingual features. Moreover, the N-gram content-based method outperforms other content-based methods. However, the lingual feature based method, with lower performance than other methods, is time efficient because it deals with a few features determined by the user, but the other three methods (N-gram, TF-IDF, and LSTM) deal with the words as features; therefore, are very time consuming both for training and for testing. Furthermore, the LSTM needs a large memory volume, mainly for loading vectorized words to process them.

Comparing the results for both datasets also shows that the measured metrics for KNTUPT dataset outperform the metrics for Sepehr_RumTel01 dataset overall. The nature of both datasets could justify it. The KNTUPT posts are derived from 60 main news [45]; therefore, a post may have appeared several times with some repeating words. Thus, as expected, the methods concerning the content of news perform better than the lingual feature based method on the dataset with more similar posts. On the other hand, the lingual feature based method performs relatively the same on both datasets due to its concentration on the writing style features, not the words or the meaning carried by the posts. Thus, it is expected that a rumor whose content has previously been recognized is more effectively classified by the content-based methods, while the lingual feature based method performs better in classifying a post encountering for the first time.

On the Sepehr_RumTel01 dataset, the overall performance of WMV is very close to the best classifier, BoW. Indeed, since WMV is a weighted average voting, classifiers other than BoW negatively affect the performance of WMV compared with BoW, but since BoW has more weight, the final WMV results tend to Bow. Interestingly, the precision of WMV (0.91) is better than every base classifier, whereas the recall of WMV (0.77) is the worst one. Since the WMV is a voting classifier, the overall performance (f-score) should not be better than each of its input classifiers. Considering the results on KNTUPT, the performance parameters (P, R, F1, Acc.) of the WMV classifier are very close to the best base classifier, BoW.

Generally speaking, since the WMV is a voting classifier, the overall performance (f-score) is not better than each of the base input classifiers, but the final result is closer to the best base classifier. If news samples have various characteristics, and the best classifier is not pre-determined, but determined according to the sample features and weights assigned according to their performance on similar samples, this fusion method is more beneficial compared to a single rumor detection method.

In order to compare with similar works, unfortunately, few similar studies in the Persian language have been reported. Table VI compares the proposed model with previous works on Persian rumor detection that have used the KNTUPT and the Sepehr_RumTel01 datasets. Although our proposed model is better than some of (not all of) these works, it is notable that the main purpose of our study is to investigate the effect of fusion on rumor detection classification.

TABLE VI. Comparison of the Proposed Method with the Previous Rumor Detections on the Datasets

Dataset	Model	Results			Features		
		P	R	F1	User	Content	Context
KNTUPT	[45]	0.97	0.99	0.99	✓	✓	✓
	[36]	0.96	0.95	0.96	-	✓	-
	[35]	0.95	0.96	0.96	-	✓	-
	Proposed	0.98	0.96	0.97	-	✓	-

[DOI: 10.61186/itrc.16.4.33]

Sepehr_RumTel01	[43]	0.79	0.79	0.79	-	✓	-
	[36]	0.93	0.93	0.93	-	✓	-
	[35]	0.98	0.91	0.94	-	✓	-
	Proposed	0.91	0.77	0.83	-	✓	-

In general, in combining the results of several classifiers, simple averaging is done and the importance of each classifier is considered the same. However, since in the proposed method, each classifier is given a different weight depending on its performance; thus, the performance of the final result of combining the results is closer to the performance of the better classifier. This point is more useful when due to the different nature or characteristics of each of the news samples, the top classifiers may be different, so in general, the fusion method can have a good effect on the classification of the samples.

VI. CONCLUSION

The widespread of rumors on social media may affect society's opinion; therefore, various methods for rumor detection have been proposed in the literature. In this study, we implemented four rumor detection classifiers using lingual feature-based, word frequency-based (BoW and TF-IDF), and word embedding-based methods. We proposed a multi-classifier model to fuse the results of the four classifiers using the weighted majority voting (WMV) method whose weights are proportional to the classifiers' F1-score values.

Regarding the nature of the datasets we used and the results of applying the classification methods on two datasets, we can conclude that:

- For the previously published posts with the same or similar contents, which are in the training set for the classifier, the content-based classifiers N-gram, TF-IDF, and LSTM outperform the lingual feature based classifier, and among these three classifiers, N-gram performs better than the others.
- The lingual feature-based classifier has roughly the same performance on the posts, regardless of whether the posts (or posts similar to it) have been seen before or not.
- The fusion of the classification results based on the (WMV) method whose weights are proportional to the F1-score of individual classifiers results in a final performance very close to the best base classifier. Therefore, the proposed multi-classifier architecture is recommended to extract weights of base classifiers according to their performance on the class of input samples, then apply classification in a multi-classification architecture with weights assigned according to the input feature class, which could be considered as future work on this study.
- The training and testing time of the N-gram, TF-IDF, and LSTM is the main deficiency of these classification methods compared with the lingual feature based classifier. Furthermore, N-gram, TF-IDF, and LSTM classifiers suffer from high memory usage, especially for LSTM, which requires a large amount of memory for vectorized words.

The main limitation of this research is the lack of enough rumor datasets in Persian, both in the number of datasets and the volume of data in available datasets. Therefore, the results of this study could be verified by applying our method to more datasets. Thus, one of the future works very helpful for this study and similar studies is developing Persian rumor datasets.

This study could also be extended from other viewpoints. We considered classification methods concerning the contents of the posts, both lingual features and semantic features, while other viewpoints of rumor could also be considered, including propagation, news source, and temporal features, as well as automatic fact checking. The fusion of classifiers' results from various viewpoints and assigning weights to the base classifiers dynamically considering feature classes of input samples could result in more comprehensive and accurate results.

REFERENCES

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD explorations newsletter*, vol. 19, pp. 22-36, 2017.

[2] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 53, pp. 1-40, 2020.

[3] A. Mansouri and F. Taghiyareh, "Effect of Segregation on Opinion Formation in Scale-Free Social Networks: An Agent-based Approach," *International Journal of Engineering*, vol. 34, pp. 66-74, 2021.

[4] A. Mansouri and F. Taghiyareh, "Effect of segregation on the dynamics of noise-free social impact model of opinion formation through agent-based modeling," *International Journal of Web Research*, vol. 2, pp. 36-44, 2019.

[5] J. A. Hołyst, K. Kacperski, and F. Schweitzer, "Social impact models of opinion dynamics," *Annual Reviews Of Computational PhysicsIX*, pp. 253-273, 2001.

[6] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch, "Mixing beliefs among interacting agents," *Advances in Complex Systems*, p. 11, 2001.

[7] A. Mansouri and F. Taghiyareh, "Toward an emotional opinion formation model through agent-based modeling," in *2017 7th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2017, pp. 14-19.

[8] A. Mansouri and F. Taghiyareh, "Phase transition in the social impact model of opinion formation in scale-free networks: The social power effect," *Journal of Artificial Societies and Social Simulation*, vol. 23, 2020.

[9] A. Mansouri and F. Taghiyareh, "Phase Transition in the Social Impact Model of Opinion Formation in Log-Normal Networks," *Journal of Information Systems and Telecommunication (JIST)*, vol. 1, p. 1, 2021.

[10] S. Bradshaw and P. N. Howard, "The global disinformation order: 2019 global inventory of organised social media manipulation," 2019.

[11] E. Chen, H. Chang, A. Rao, K. Lerman, G. Cowan, and E. Ferrara, "COVID-19 misinformation and the 2020 US presidential election," *The Harvard Kennedy School Misinformation Review*, 2021.

[12] J. P. Baptista and A. Gradim, "Online disinformation on Facebook: the spread of fake news during the Portuguese 2019 election," *Journal of Contemporary European Studies*, pp. 1-16, 2020.

[13] S. T. Smith, E. K. Kao, E. D. Mackin, D. C. Shah, O. Simek, and D. B. Rubin, "Automatic detection of influential actors in disinformation networks," *Proceedings of the National Academy of Sciences*, vol. 118, 2021.

[14] N. Grinberg, K. Joseph, L. Friedland, B. Swire-Thompson, and D. Lazer, "Fake news on Twitter during the 2016 US presidential election," *Science*, vol. 363, pp. 374-378, 2019.

[DOI: 10.61186/itrc.16.4.33]

[15] A. Bovet and H. A. Makse, "Influence of fake news in Twitter during the 2016 US presidential election," *Nature communications*, vol. 10, pp. 1-14, 2019.

[16] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *Journal of economic perspectives*, vol. 31, pp. 211-36, 2017.

[17] M. T. Bastos and D. Mercea, "The Brexit botnet and user-generated hyperpartisan news," *Social science computer review*, vol. 37, pp. 38-54, 2019.

[18] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, pp. 1-36, 2018.

[19] Z. Jahanbakhsh-Nagadeh, M.-R. Feizi-Derakhshi, and A. Sharifi, "A Model for Detecting of Persian Rumors based on the Analysis of Contextual Features in the Content of Social Networks," *Signal and Data Processing*, vol. 18, pp. 50-29, 2021.

[20] M. H. Saghayan, S. F. Ebrahimi, and M. Bahrani, "Exploring the Impact of Machine Translation on Fake News Detection: A Case Study on Persian Tweets about COVID-19," in *2021 29th Iranian Conference on Electrical Engineering (ICEE)*, 2021, pp. 540-544.

[21] S. Zamani, M. Asadpour, and D. Moazzami, "Rumor detection for persian tweets," in *2017 Iranian Conference on Electrical Engineering (ICEE)*, 2017, pp. 1532-1536.

[22] L. Zhou, J. K. Burgoon, J. F. Nunamaker, and D. Twitchell, "Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications," *Group decision and negotiation*, vol. 13, pp. 81-106, 2004.

[23] C. M. Fuller, D. P. Biros, and R. L. Wilson, "Decision support for determining veracity via linguistic-based cues," *Decision Support Systems*, vol. 46, pp. 695-703, 2009.

[24] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *2012 IEEE Symposium on Security and Privacy*, 2012, pp. 461-475.

[25] M. Siering, J.-A. Koch, and A. V. Deokar, "Detecting fraudulent behavior on crowdfunding platforms: The role of linguistic and content-based cues in static and dynamic contexts," *Journal of Management Information Systems*, vol. 33, pp. 421-455, 2016.

[26] D. Zhang, L. Zhou, J. L. Kehoe, and I. Y. Kilic, "What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fake online reviews," *Journal of Management Information Systems*, vol. 33, pp. 456-481, 2016.

[27] G. D. Bond, R. D. Holman, J. A. L. Eggert, L. F. Speller, O. N. Garcia, S. C. Mejia, *et al.*, "'Lyin'Ted','Crooked Hillary', and 'Deceptive Donald': Language of Lies in the 2016 US Presidential Debates," *Applied Cognitive Psychology*, vol. 31, pp. 668-677, 2017.

[28] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," *arXiv preprint arXiv:1702.05638*, 2017.

[29] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," *arXiv preprint arXiv:1708.07104*, 2017.

[30] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani, "Fake news early detection: A theory-driven model," *Digital Threats: Research and Practice*, vol. 1, pp. 1-25, 2020.

[31] B. Al Asaad and M. Erascu, "A tool for fake news detection," in *2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2018, pp. 379-386.

[32] H. E. Wynne and Z. Z. Wint, "Content based fake news detection using n-gram models," in *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, 2019, pp. 669-673.

[33] M. Samadi, M. Mousavian, and S. Momtazi, "Persian fake news detection: Neural representation and classification at word and text levels," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, pp. 1-11, 2021.

[34] M. Samadi and S. Momtazi, " Fake news detection: deep semantic representation with enhanced feature engineering," *International Journal of Data Science and Analytics*, pp. 1-12, 2023.

[35] Z. Jahanbakhsh-Nagadeh, M.-R. Feizi-Derakhshi, and A. Sharifi, "A Deep Content-Based Model for Persian Rumor Verification," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, pp. 1-29, 2021.

[36] Z. Jahanbakhsh-Nagadeh, M.-R. Feizi-Derakhshi, and A. Sharifi, "A semi-supervised model for Persian rumor verification based on content information," *Multimedia Tools and Applications*, vol. 80, pp. 35267-35295, 2021.

[37] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "Parsbert: Transformer-based model for persian language understanding," *Neural Processing Letters*, vol. 53, pp. 3831-3847, 2021.

[38] V. Mottaghi, M. Esmaeili, G. A. Bazaee, and M. Afshar Kazemi, "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*, vol. 8, pp. 1-17, 2021.

[39] [M. Ghayoomi and M. Mousavian, "Deep transfer learning for COVID-19 fake news detection in Persian," *Expert Systems*, p. e13008, 2022.

[40] M. M. Sadr, "The Use of LSTM Neural Network to Detect Fake News on Persian Twitter," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, pp. 6658-6668, 2021.

[41] A. Rahman and S. Tasnim, "Ensemble classifiers and their applications: A review," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 10, pp. 31-35, 2014.

[42] L. Hasimi and A. Poniszewska-Marańda, "Ensemble Learning-based Fake News and Disinformation Detection System," in *2021 IEEE International Conference on Services Computing (SCC)*, 2021, pp. 145-153.

[43] Z. Jahanbakhsh-Nagadeh, M.-R. Feizi-Derakhshi, and A. Sharifi, "A speech act classifier for persian texts and its application in identifying rumors," *Journal of Soft Computing and Information Technology*, vol. 9, pp. 18-27, 2020.

[44] A. R. Feizi Derakhshi, M. R. Feizi Derakhshi, M. Ranjbar-Khadivi, N. Nikzad Khasmakhi, M. Ramezani, T. Rahkar Farshi, *et al.*, "'Sepehr_RumTel01", Mendeley Data, doi: 10.17632/jw3zwf8rdp," ed, 2020.

[45] S. D. Mahmoodabad, S. Farzi, and D. B. Bakhtiarvand, "Persian rumor detection on twitter," in *2018 9th International Symposium on Telecommunications (IST)*, 2018, pp. 597-602.

[46] S. Singhal, A. Kabra, M. Sharma, R. R. Shah, T. Chakraborty, and P. Kumaraguru, "Spotfake+: A multimodal framework for fake news detection via transfer learning (student abstract)," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 13915-13916.

Alireza Mansouri is an Assistant Professor of ICT Research Institute (ITRC). He received his BSc and MSc from Sharif University of Technology, both in Computer Engineering-Software and his PhD in Computer Engineering-Information Technology from University of Tehran. He is currently deputy of "IT faculty" and head of "Data Analysis Research Group". His research interests include data science, computational social science, cloud computing, and agent-based modeling and simulation.

[DOI: 10.61186/itrc.16.4.33]

Maryam Mahmoudi holds a B.Sc. in Software Engineering and a M.Sc. in Information Technology Engineering. She has been a researcher at the ICT Research Institute since 2012. Her areas of expertise include Information Retrieval, Data Mining, Natural Language Processing, Artificial Intelligence, and Generative AI.

Mojgan Farhoodi received her B.Sc. degree in Software Engineering and her M.Sc. and Ph.D. degree in IT Engineering and IT management Respectively. She has been working as a researcher at the ICT Research Institute since 2010. Currently, she is head of AI lab and faculty member of the ICT research institute. Her areas of expertise are Information Retrieval, Data Mining, Natural Language Processing, and Artificial Intelligence.

Mohammadreza Mirsarraf received his Ph.D. degree in telecommunications in 2001. His main research interest is cloud computing, Internet of Things, real-time operating systems, fifth generation wireless networks, and next-generation networks. He is also interested in using semiology and pragmatism theory on human–computer interaction design.

[DOI: 10.61186/itrc.16.4.33]