Rumor Detection on Twitter using Extracted Patterns from Conversational Tree

Arefeh Yavary
Department of Computer Science, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran
yavary_rf@ut.ac.ir

Hedieh Sajedi
Department of Computer Science, School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran
hhsajedi@ut.ac.ir

***Abstract*— These days, Twitter social network is one of the main platforms for getting news, among people over the world. This is because of the high volume of data generated by this social media, which makes Twitter up to date with news and information. Nevertheless, the existence of invalid information over the social network makes the users unhappy and also arises some problems in the real world, particularly in the crisis. To overcome these problems and other possible issues, automatic detection of rumor on Twitter must be taken into account. Despite such issues, in this paper, rumor detection in Twitter is studied. In this paper, the rumor is validated by considering the user's feedback, as the source data for rumor study. In our proposed method, pattern recognition and its analysis of the user conversational tree in Twitter is studied. These recognized patterns feed into as features for training a classifier for rumor detection. The model for training a classifier is an Extreme Learning Machine and its extension. The dataset for experiments of our method is the standard dataset of SemEval-2017 Task 8. Experiments of our proposed method with respect to competitor methods in rumor detection show that our method outperforms the state of the art methods.**

Keywords*— *Rumor Detection; Twitter; Social Network; Conversational Tree; Pattern Recognition; Extreme Learning Machine

I. INTRODUCTION

The social network is widely used by people over the world these days. This global use leads to the high volume of information generated over the social network that turns the social network into an appropriate alternative for the news agency [1]. Although, we know that the accessibility and ease in use of social networks make it befitting for rumor dispersion too. This rumor dispersion could cause worry and concern among people, especially in the critical time and situations. By this, automatic detection of rumor messages and posts on the social network is very necessary. Among the existing social networks, Twitter, Sina Weibo (a Chinese microblogging website) and Facebook are the frequently used ones over the world [2]. Among these three, Twitter is chiefly used for receiving up to date information and news [3]. Because of the fact that the Tweet has the special properties which make it apt for rapid propagation among people and over the social network [3].

By the reasons stated in the previous paragraph, our research goal of this paper is the study of rumors in the Twitter social network in order to classify tweets based on their validity. In this task, the level of validity of tweets is considered by three classes of ‘true’, ‘false’ and ‘unverified’.

Howbeit, we study the rumors of the Twitter, in the previously related researches, other social networks like as Facebook [4-6] and Sine Weibo [7-11] are examined too.

Also, in the higher level of grouping previous literature on the subject of rumor study in cyberspace, independent of the kind of the social network framework, dependent pieces of literature are divided into two cases of rumor detection [12-13] and rumor diffusion prevention [14-16]. In this research, rumor detection is our goal study.

The related work on the subject of rumor detection on the social network shows that most of the researches have been employed in investigating the structure of social network [5] or study the content of the examined rumor [13]. But also, the feedback of the users is not studied properly in rumor detection, which could be useful in the quick detection of rumors [17].

However, as the social network content is created by the users, the feedback from the users to the posted items is crucial and important in rumor detection. By this fact, in this study, we consider the user feedback as the main and a crucial source for rumor detection.

In favor of examining and study of rumor feedback in rumor detection, we benefited from the user conversational tree of the examined tweet. The user conversational tree is described as the tree of user’s reply in response to the case study tweet. Also, each reply is annotated with one of the tags of support, deny, comment or query. These tags are applied in order to specify the reaction of others to the analyzed tweet for rumor detection in a measurable form.

The proposed analysis of user conversational tree draws on the recognized patterns on the conversational tree. These

978-1-5386-5364-7/18/$31.00 ©2018 IEEE

extracted patterns are divided into two groups: patterns based on conversational tree and patterns based on sub-trees. The former group of patterns is like as n-grams on the tags of replies and the latter group of patterns is small subtrees with different sizes. Then the extracted patterns are used as features for Extreme Learning Machine (ELM) [18-19] and the model for rumor detection is constructed. In our model construction, ELM is used in two cases of basic and kernel extensions.

ELM [18, 19] is suggested in order to train single layer feedforward neural networks (SLFNs). This training method is very fast and giving very satisfiable results in different applications [20]. In addition, ELM has been developed in various extensions and also in combination with other methods for several usages.

Proper and accessible datasets for our research topic is so rare. As we know, the only publicly available dataset for rumor detection is what mentioned in [17]. The experiments of our proposed method, are performed on this dataset. The baselines of experiments [17] are the state of the art methods in SemEval-2017 Task 8. Experimental results show that our proposed method outperforms the state of the art methods.

In the following of this paper, first, the literature review of researches on rumor detection and also the ELM method is studied. Then, primary knowledge for ELM and Kernel-ELM is explained. After that, the proposed approach of this paper is discussed. This section contains the explanation of method in overall, including pre-process and pattern extraction parts of the proposed method. Experiments and analysis of our suggested method are explained afterward. At the end, conclusion and future works are presented.

II. Literature Review

In this part, related work on rumor study, especially rumor detection, as the target of this study is reviewed. Furthermore, as ELM is selected as a classification method in our proposed method, the applications of ELM in different research work are studied, briefly.

A. Rumor Study

Research on the topic of rumor study is presented in two cases of rumor detection and rumor diffusion. As our research goal is in on the case of rumor detection, the literature review in the rumor detection subsection is discussed in more details. The rumor detection literature review is considering the frameworks of social network and web. Then the social network frameworks are studied in cases of Twitter, Facebook and Sina Weibo.

1. Rumor Detection

Rumor detection is studied in different frameworks, which are mainly including social networks [4-13, 21-24] and web [25-26]. In a case study of rumor detection in social networks, Twitter [12, 13, 21-24], Sina Weibo [7-11] and Facebook [4-6] are the most studied ones, that discussed in the following.

In this part, rumor veracity prediction on Twitter is studied. Giasemidis et al. [21] over 80 trustworthiness measures in different approaches are employed to use machine learning methods to determine trustworthiness scores for rumor detection at various time windows. Furthermore, they demonstrated some attributes of the data which effects more accurately in trustworthiness scores. They set-out from their findings that their proposed model is significantly more accurate than similar studies. Also, a visualization tool for visualizing the results of the proposed model and further analyze is developed, too. Qin et al. [22], they suggest the novelty-based features that are not considered in the previous system. These previous system's features are repetitive and result in the slow detection of rumors. They also admitted a pseudo feedback feature, which is using the previous rumor documents. This feature with the combination of novelty-based features shows the best result in the performance of the rumor detection. Zubiaga et al. [12], the validation of news is considered. The employed method that aims to use the sequential dynamics of news. They used model is a CRF (Conditional Random Field) which perform better than similar studies in this task. Zubiaga et al. [13], an algorithm for rumor source detection is suggested. This algorithm is verified on a network with different sizes of 1 K, 10 K, and 100 K. The results demonstrate that the network with more nodes gives the better results. Dayani et al. [24], different machine learning methods like as Naive Bayes or k-nearest neighbor classifier employed for rumor detection.

Up to this, information veracity is studying independent of information topic. But Sicilia et al. [23], information veracity is studied under a health topic. Their method is considering features in influencing potential and network characteristics.

The next social network service for the literature review is Sina Weibo. The number of users is this social network is more than eight numbers of times of the users on Twitter. Yang et al. [7], a classifier for rumor detection based on the social network's users is trained for rumor classification. Zhou et al. [8], the first real-time news certification system is created by an ensemble model. This ensemble model is used on user-based, propagation-based, and content-based models. Wu et al. [9], a graph-kernel based hybrid SVM classifier is proposed for rumor detection. This model is considering propagation patterns and also semantic features like as topics and sentiments. Against traditional methods for rumor detection, which use handcrafted features, Ma et al. [10] a method for automatic learning of continuous representations for rumor detection is presented. This model used recurrent neural networks (RNN). Liu et al. [11], a diffusion process for Sina Weibo is studied in order to utilize diffusion features for creating rumor classifier.

In the case of rumor detection on the web, not so much study done. In this review, two cases are discussed. The first one is based on Wikipedia [25] and the other one is exploring the entire web [26]. Kumar et al. [25], the goal study is to detect invalid articles and information in Wikipedia. This work is valuable because of the fact that Wikipedia is the source of many general knowledge bases, these days. For example, Derczynski et al. [17], one of the Wikipedia dumps is considered as the knowledge base for the open variant veracity prediction task [17] in Semeval-2017 task 8. Open variant veracity prediction means that the information further than which could get from the social networks of the examined tweet could be used for information credibility. The study at [25], tries to detect false rumors by recognizing hoax articles (articles containing fabricated facts about non-existent entities or events) in Wikipedia. In addition, humans are not good at recognizing hoax articles [25] and by this, hoax detecting articles is very helpful. This paper transforms hoax article detection to a classification problem and shows that the suggested classifier outperforms the other methods.

Galitsky et al. [26], the validation of a given information is measured by comparing two texts by hybrid means of web mining and linguistic-based methods. The proposed algorithm is evaluated on a customer's review dataset of the product reviews. The experimental results show the befit of the proposed algorithm for misinformation detection.

2. Rumor Diffusion

In this section, rumor diffusion studies, by attention to the special case of the Twitter platform are studied.

Okada et al. [15], an information diffusion model for considering invalid rumor on Twitter is suggested. This model is an extension of SIR (susceptible, infectious and recovered) model. Also, a strategy for prevention of false rumor diffusion in Twitter is shown in this paper. This information diffusion model has one peak over its model, and also some constraints and properties for users with different numbers of follow/follower in prevention strategy of false rumor spread. This research emphasizes that other social network services must be studied separately for getting its information diffusion model.

Malkhi et al. [16], a group of diffusion algorithms for false rumor spreading prevention is designed. The performance of these algorithms is optimal in diffusion measures.

Takayasu et al. [14], rumor diffusion and its convergence in Twitter in emergency conditions are studied. In modeling of diffusion, stochastic agent-based model, which is an extension of SIR model is considered.

B. Extreme Learning Machine

The proper papers for defining ELM and its learning strategies are [18, 19]. In [18] basic ELM and its belonging, learning theory is defined and in [19], ELM in Kernel extension is defined.

ELM and its extension methods are employed for several applications with fascinating results. Also Yahia et al. [20], ELM and widely used wavelet neural network are compared in a classification task using several datasets. Furthermore, some novel learning methods are recently developed been prepared on the basis of ELM [27] such as FI-ELM and EI-ELM. FI-ELM is the faster version of EI-ELM. EI-ELM is the learning method that aims to avoid the problem of the minor effects of some hidden neurons in ELM by an enhanced random search based on an incremental extreme learning machine. Tang et al. [28] a novel hierarchical learning framework based on ELM is proposed for multilayer perceptron. Song et al. [29], a multi-resolution selective ensemble ELM learning method for time-series prediction is suggested. This learning method is applied for prediction of next-step and next-day electricity consumption.

III. Primary Knowledge

The necessary primary and background knowledge for the description of our proposed method is an ELM model in basic and its kernel case which are described in the following section.

A. Extreme Learning Machine

ELM [17] is a single hidden layer feed-forward neuronal network. The property that makes the ELM special is its magical learning method. The structure of ELM is illustrated in Fig. 1. In this figure, X (with size N) as input is given to the ELM network. In classification application of ELM, the tag of the input gives as output. As you can see in Fig. 1, the ELM is constructed of three layers: input layer, a hidden layer, and output layer.

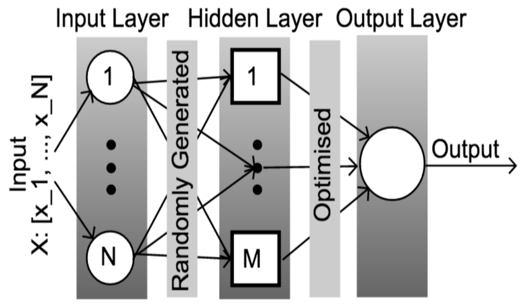

Fig. 1. Structure of ELM. X is the input, N is the input length and M is the size of the hidden Layer.

The learning strategy of ELM is at this: weights of edges between the input layer and hidden layers are randomly generated, and then the weights of layers between the hidden layer and output layer are optimized. The mathematical procedure of this method is coming in the following description.

Consider the set $\mathcal{R} = \{(x_i, t_i) \mid xi \in R_n, \text{the} \in R_m, i = 1,\dots, N\}$ as train set with N samples, g (x) as activation function and Ñ as the number of nodes in the hidden layer. We define matrix T and β as (1) and (2), respectively:

$$T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{N}}^T \end{bmatrix}_{N \times m} \tag{1}$$

In (1), *T* is the matrix of input data tags.

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix} \tag{2}$$

In (2), β is the weights of edges between hidden layer and output layer in ELM network, which we aim to optimize it.

ELM learning strategy is as follows:

1. Assign input weight w_i and bias bi (for i = {1… Ñ}), randomly.
2. Calculate matrix H of hidden layer by considering the activation function g by the computation that formulated in (3).

$$H\left(w_1,\ldots,w_{\tilde{N}},b_1,\ldots,b_{\tilde{N}},x_1,\ldots,x_N\right) = \begin{bmatrix} g\left(w_1.x_1+b_1\right) & \cdots & g\left(w_{\tilde{N}}.x_1+b_{\tilde{N}}\right) \\ \vdots & \cdots & \vdots \\ g\left(w_1.x_N+b_1\right) & \cdots & g\left(w_{\tilde{N}}.x_N+b_{\tilde{N}}\right) \end{bmatrix}_{N\times\tilde{N}} \quad (3)$$

1. Then Moore–Penrose generalized inverse of matrix H, must be computed. This kind of inverse computation is noted by H†. For more information about $H^{\dagger}$, you can refer to [30, 31].
2. After that, matrix β of weights can be computed as follows:

$$\beta = H^{\dagger}T \quad (4)$$

By computing β, the ELM network is ready to use.

The activation functions employed in basic ELM are Sigmoidal function, Sine function, Hardlim function, Triangular basis function and Radial basis function. In the following, each mathematical definition of the mentioned activation functions is defined [19]:

• Sigmoidal function: The sigmoidal function is given as the following equation by *f (x, a, c),* which is a mapping on a vector x, and depends on the *a* and *c* parameters.

$$f\left(x,a,c\right)=\frac{1}{1+e^{-a(x-c)}} \quad (5)$$

• Sine function: As (6), Sine function is the popular sin trigonometric function.

$$Sine\left(x\right) = Sin\left(x\right) \quad (6)$$

• Hardlim function: Hardlim (Hard Limit transfer function) is defined as follows:

$$hardlim\left(n\right)=\begin{cases}1 & n\geq 0\\ 0 & otherwise\end{cases} \quad (7)$$

The Hardlim output is 1, if n is less than or equal to 0, and otherwise equals to 0.

• Triangular basis function: Tribas is a neural transfer function, which is defined in (8).

$$tribas\left(n\right)=\begin{cases}1-abs\left(n\right) & -1\leq n\leq 1\\ 0 & otherwise\end{cases} \quad (8)$$

abs(n) returns, absolute value of parameter *n*.

• Radial basis function: Radial basis function which is represented as *radbas(n)*, as defined in (9).

$$radbas\left(n\right)=exp\left(-n^2\right) \quad (9)$$

B. Kernel-based Extreme Learning Machine

Several classes of algorithms can benefit from kernel methods using kernel functions and ELM is a member of this class of algorithms. Kernel functionality in ELM is defined as the common use of kernel method, which considering the unknown feature mapping φ by the kernel function *k*. The kernel function is defined as (10):

$$k\ \left(u,\ v\right)\ =\ \varphi(u).\ \varphi(v) \quad (10)$$

Like as basic ELM, ELM learning method using the kernel, get input *X* and the output like as the basic ELM. The set $\mathscr{R}$ = $\{(x_i, t_i) \mid x_i \in R^n,\ t_i \in R^m,\ i=1 \ldots N\}$ is considered as a train set with *N* samples and *g(x)* as activation function. Also, matrix *T* and β are defined same as (1) and (2), with the same role.

ELM-kernel based learning steps are as follows:

Step 1 and 2: Do Step 1 and 2 same as what explained for ELM in basic mode.

Step 3. For computing β do like as (11):

$$\beta = H^T\left(\frac{I}{C}+HH^T\right)^{-1}T \quad (11)$$

In (11), *C* is a user-specified parameter. This parameter is defined to handle a trade-off between minimizing the training error and maximizing the distance $2/||\beta||$ of the splitting margin of the ELM feature space.

Step 4. Define a kernel matrix for ELM as (12):

$$\Omega_{ELM} = HH^T : \Omega_{ELM_{i,j}} = h\left(x_i\right).h\left(x_j\right)=K\left(x_i,x_j\right) \quad (12)$$

Step 5. Compute ELM output function *f(x)* by (13), where *h(x)* is the feature mapping function:

$$f(x)=h(x)H^{T}\left(\frac{I}{C}+HH^{T}\right)^{-1}T=\begin{bmatrix}K(x,x_1)\\ \cdot\\ \cdot\\ \cdot\\ K(x,x_N)\end{bmatrix}^{T}\left(\frac{I}{C}+\Omega_{ELM}\right)^{-1}T \tag{13}$$

The kernel-based ELM method has several kernel functions. The kernel functions that used in this paper are: RBF Kernel, Linear Kernel, Polynomial Kernel and Wavelet Kernel. These Kernels are defined as follows [32]:

• RBF Kernel: The RBF (Radial Basis Function) kernel that popular as Gaussian kernel function, used in various kernel-based learning algorithms. The RBF kernel on two samples *x* and *y*, represented as feature vectors in some input space, is defined as:

$$K(x,y)=exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{14}$$

In (14), $||x\text{-}y||^2$ is the squared Euclidean distance between the *x* and *y* feature vectors and σ is a constant term parameter.

• Linear Kernel: Linear Kernel is defined as follows, which c is a free parameter:

$$K(x,y)=x^{T}y+c \tag{15}$$

• Polynomial Kernel: For polynomials with polynomial degree *d*, the polynomial kernel function is defined (16). The slope α and *c* are parameters that must be adjusted.

$$K(x,y)=(\alpha x^{T}y+c)^{d} \tag{16}$$

IV. PROPOSED APPROACH

This section begins with the description of the overall process of our approach and continued with the description of its sub-processes in details. The sub-process, including pre-processing, pattern extraction on user conversational tree and modeling sections.

A. *Overall Process*

The overall process containing three main parts: 1. Preprocess, 2. Pattern Extraction, 3. Modeling. The overall process is illustrated in Fig. 2. In this image, the three main elements of this process with their corresponding sub-process are shown.

In the first step of the overall process, the tagged user conversational tree is given as an input to the preprocessing phase. The preprocess phase, including two sub-process of branch extraction and level set extraction of three. In step 2, pattern extraction is done. Pattern extraction step is including two sub-process of n-gram and branched sub-trees pattern extraction.

B. *Pre-processing*

The task of pre-processing of the user conversational tree contains two following sub-process: 1. Branch Extraction and 2. Level Extraction.

By a combination of these two sub-processes, the pattern extraction is done. Each of the two sub-processes is explained in the continued subsections:

1. Branch Extraction

This preprocessing step is coming to account in order to extract branches of the user conversational tree for n-gram pattern extraction. The Fig. 3.a. is demonstrated the task of branch extraction.

2. Level *Extraction*

The level extraction task is used for subtree pattern extraction. In this preprocessing step, each node with the same level is put to the same level-set. This action of this sub-process is illustrated in Fig. 3.b.

D. *Extracted Patterns*

Extracted patterns of user conversational tree are split into two groups: n-gram patterns and branched sub-tree patterns. These patterns are illustrated in Fig. 3. In this figure, some samples of these patterns are shown. The n-gram patterns, just consider the branches of the conversational tree and ignore the dependency of branches, because of that, the branched sub-tree patterns introduced in order to study the dependency between branches of the conversational tree. In the two following subsections of this part, these two patterns are explained.

1. N-gram patterns

N-gram patterns are the contiguous sequence of *N* items. By this definition, *N* is the length of the pattern. For example,

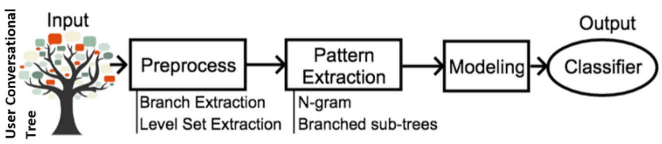

Fig. 2. Overall process of the proposed approach. This process involves three main parts, which are preprocessing, pattern extraction, and modeling.

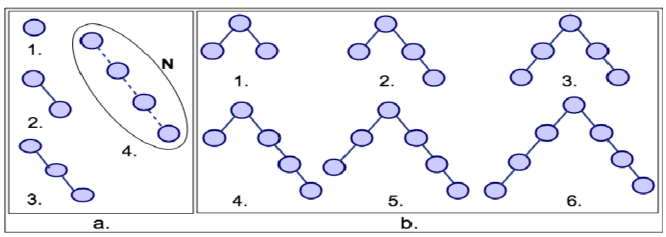

Fig. 3. . Extracted patterns from user conversational tree.
(a) and (b) show the N-gram patterns and branched-patterns of user conversational tree, respectively.

pattern *"deny-deny-support-query"* is the pattern with length 4, thus this is a 4-gram. Different N-gram patterns with different length are extracted to find the proper length of N-gram patterns. In Fig. 3.a. the patterns of unigram (1-gram), bigram (2-gram), 3-gram, and N-gram are illustrated. These patterns are extracted using the branch extraction pre-processing step

1. Branched sub-tree patterns

These kinds of patterns are defined in order to consider the relation of the branches of the user conversational tree, which are not considered in the n-gram patterns. These patterns are extracted using the branch extraction and level extraction pre-processing step. In Fig. 3.b., these patterns are illustrated.

V. EXPERIMENTS

In this section, the consideration of the experiments, such as setups of experiment environments, methods of comparisons and results are described.

A. Experiment Environment Setups

- Dataset

The data set is a publicly available data set, which is used in SemEval-2017 Task 8 [17]. This dataset is the only publicly available dataset for rumor detection and because of that, the experiments are done just on this dataset. This dataset contains 247 tweets in the training set (137 true, 62 false, 98 unverified) and 28 tweets in the test set (8 true, 12 false, 8 unverified).

- Computational Environment

The preprocessing is implemented in Python 3.6. ELM and its kernel extension are implemented in Matlab-R2016-a. Other machine learning classification algorithms are implemented using Scikit-Learn library in Python.

B. Methods of comparison

The previous methods are tested on the dataset of [17] and they employed popular and common classification methods.

C. Results

In this part, the results on the train and test set with the comparison methods and baseline systems, which were defined in [17], are presented.

The baselines are the systems, which defined in [17]. The results on the unigram, bigram, 3-gram, and 4-gram patterns are tested. But, as the results on bigram to 4-gram are the same. Therefore, just results on unigram and 4-gram are presented. In TABLE I and TABLE II, the results for train and test on unigram and in TABLE III and TABLE IV, for train and test on 4-gram are presented, respectively. The evaluation measures that defined in TABLE I to IV, are Score, Confidence RMSE, and Final Score. These are same as which defined in [17]. Score is the same as accuracy. Confidence RMSE is a new measure of confidence which defined for rumor detection, which defined in [17]. Final Score is defined as *Score×(1-Confidence RMSE)*. In all the tables, bold numbers illustrate the best-obtained results.

In Table I, three evaluation measures are listed for all of the proposed approaches and methods for comparison. These results are for a train set. The pattern of study is unigram.

TABLE I. Experimental results on the train set for unigram.

Approach	Evaluation Measures		
	Score	***Confidence RMSE***	***Final Score***
Elm-kernel (RBF Kernel)	*68.4%*	*0.478*	*0.357*
Elm-kernel (Linear Kernel)	*46.7%*	*0.846*	*0.072*
Elm-sig	*97.1%*	*0.037*	*0.935*
Elm-hardlim	*96.0%*	*0.063*	*0.900*
Elm-sine	*97.1%*	*0.037*	*0.935*
Elm-rbfs	*96.0%*	*0.063*	*0.900*
Elm-tribas	*97.1%*	*0.037*	*0.935*
Multinominal Naive Bayes	*46.3%*	*0.537*	*0.215*
Support Vector Machine	*82.0%*	*0.180*	*0.672*
Multi-Layer Perceptron	*18.4%*	*0.816*	*0.034*

TABLE II. EXPERIMENTAL RESULTS ON THE TEST SET FOR UNIGRAM

Approach	Evaluation Measures		
	Score	***Confidence RMSE***	***Final Score***
Elm-kernel (RBF Kernel)	*50.0%*	*0.679*	*0.161*
Elm-kernel (Linear Kernel)	*25.0%*	*1.036*	*-0.009*
Elm-sig	***60.7%***	***0.536***	***0.282***
Elm-hardlim	*53.6%*	*0.607*	*0.210*
Elm-sine	*57.1%*	*0.643*	*0.204*
Elm-rbfs	*53.6%*	*0.607*	*0.210*
Elm-tribas	*57.1%*	*0.607*	*0.224*
Multinominal Naive Bayes	*42.9%*	*0.571*	*0.184*
Support Vector Machine	*32.1%*	*0.679*	*0.103*
Multi-Layer Perceptron	*42.9%*	*0.571*	*0.184*
DFKI DKT	*39.3%*	*0.845*	*0.061*
ECNU	*46.4%*	*0.736*	*0.122*
IITP	*28.6%*	*0.807*	*0.055*
IKM	*53.6%*	*0.736*	*0.142*
NileTMRG	*53.6%*	*0.672*	*0.176*
Baseline	*57.1%*	-	-

TABLE III. Experimental results on the train set for 4-gram.

Approach	Evaluation Measures		
	Score	***Confidence RMSE***	***Final Score***
Elm-kernel (RBF Kernel)	*96.0%*	*0.063*	*0.900*
Elm-kernel (Linear Kernel)	*46.7%*	*0.846*	*0.072*
Elm-sine	*97.1%*	*0.037*	*0.935*
Elm-rbfs	*96.0%*	*0.063*	*0.900*
Elm-sine	*97.1%*	*0.037*	*0.935*
Elm-rbfs	*96.0%*	*0.063*	*0.900*
Elm-tribas	*97.1%*	*0.037*	*0.935*
Multinominal Naive Bayes	*68.4%*	*0.478*	*0.357*
Support Vector Machine	*82.0%*	*0.180*	*0.672*
Multi-Layer Perceptron	*18.4%*	*0.816*	*0.034*

In Table III, three evaluation measures are listed for all of the proposed approaches and methods for comparison. The five highlighted rows are previous methods and the last row is the baseline. These results are in the test set. The pattern of study is unigram.

In Table III, three evaluation measures are listed for all of the proposed approaches and methods for comparison. The pattern of study is 4-gram.

In Table IV, three evaluation measures are listed for all of the proposed approaches and methods for comparison. The five highlighted rows are previous methods and the last row is the baseline. These results are in the test set. The pattern of study is 4-gram.

TABLE IV. Experimental results on the test set for 4-gram

Approach	Evaluation Measures		
	Score	*Confidence RMSE*	*Final Score*
Elm-kernel (RBF Kernel)	*53.6 %*	*0.607*	*0.210*
Elm-kernel (Linear Kernel)	*32.1 %*	*0.679*	*0.103*
Elm-sig	***64.3 %***	***0.301***	***0.424***
Elm-hardlim	*60.7 %*	*0.536*	*0.282*
Elm-sine	***64.3 %***	***0.301***	***0.424***
Elm-rbfs	*60.7 %*	*0.536*	*0.282*
Elm-tribas	***64.3 %***	***0.301***	***0.424***
Multinominal Naive Bayes	*50.0 %*	*0.679*	*0.161*
Support Vector Machine	*42.9 %*	*0.571*	*0.184*
Multi-Layer Perceptron	*50.0 %*	*0.679*	*0.161*
DFKI DKT	*39.3 %*	*0.845*	*0.061*
ECNU	*46.4 %*	*0.736*	*0.122*
IITP	*28.6 %*	*0.807*	*0.055*
IKM	*53.6 %*	*0.736*	*0.142*
NileTMRG	*53.6 %*	*0.672*	*0.176*
Baseline	*57.1 %*	-	-

D. Discussion

The results on different measures show the same results for bigram, 3-gram, and 4-gram. This shows that bigram is enough to study the N-gram patterns of user conversational tree. In addition, by the use of ELM method, unigram pattern is outperforming the baseline systems in [17] and bigram pattern outperforms the unigram method in all evaluating measures.

VI. Conclusion and Future Works

In this paper, a method for rumor detection on the Twitter social network is proposed. This method is built on pattern recognition of user conversational tree and using ELM classifier. The extracted patterns are organized into two categories: 1. N-grams patterns, 2. Branched_Subtree patterns. The main groups of patterns extracted from user conversational tree branches and the succeeding ones derived from the entire conversational tree. Then the extracted patterns analyzed based upon their importance to the structure of the tree and also in a classification task of tweets validity. These patterns fed into the ELM to construct an appropriate rumor classification model based on the tweet validity. Experiments of the proposed classification model are implemented on a publicly available dataset of SemEval-2017 Task 8. These experimental results show that our suggested method is superior to other states of the art methods, proposed for this task. In future work, other distinctive patterns on the user conversational tree will be studied. In addition, these extracted patterns could be accompanied by other methods and in combination with the content and network-based features of rumor detection.

References

[1] A. Ju, S. H. Jeong, and H. I. Chyi, "Will Social Media Save Newspapers?," Journalism Practice, vol. 8, no. 1, pp. 1–17, Apr. 2013.

[2] M. Sparkes, "The ten most popular social networking websites," The Telegraph, 23-Aug-2011. [Online]. Available: http://www.telegraph.co.uk/technology/google/8718580/The-ten-most-popular-social-networking-websites.html. [Accessed: 16-Mar-2018].

[3] S. Petrovic, M. Osborne, R. McCreadie, C. Macdonald, I. Ounis, and L. Shrimpton, "Can Twitter replace newswire for breaking news?," International AAAI Conference on Web and Social Media, Jul. 2013.

[4] C. W. Tan, P.-D. Yu, C.-K. Lai, W. Zhang, and H.-L. Fu, "Optimal detection of influential spreaders in online social networks," 2016 Annual Conference on Information Science and Systems (CISS), Mar. 2016.

[5] H. Wang, P. Zhang, L. Chen, H. Liu, and C. Zhang, "Online diffusion source detection in social networks," 2015 International Joint Conference on Neural Networks (IJCNN), Jul. 2015.

[6] J. Choi, S. Moon, J. Woo, K. Son, J. Shin, and Y. Yi, "Rumor source detection under querying with untruthful answers," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, May 2017.

[7] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on Sina Weibo," Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics - MDS 12, Aug. 2012.

[8] X. Zhou, J. Cao, Z. Jin, F. Xie, Y. Su, D. Chu, X. Cao, and J. Zhang, "Real-Time News Certification System on Sina Weibo," Proceedings of the 24th International Conference on World Wide Web - WWW 15 Companion, May 2015.

[9] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on Sina Weibo by propagation structures," 2015 IEEE 31st International Conference on Data Engineering, Apr. 2015.

[10] J. Ma, W. Gao , P. Mitra , S. Kwon , B. J. Jansen, K.-F. Wong , and M. Cha , "Detecting rumors from microblogs with recurrent neural networks," IJCAI'16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, Jul. 2016.

[11] Y. Liu, X. Jin, H. Shen, and X. Cheng, "Do Rumors Diffuse Differently from Non-rumors? A Systematically Empirical Analysis in Sina Weibo for Rumor Identification," Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science, pp. 407–420, May 2017.

[12] A. Zubiaga, M. Liakata, and R. Procter, "Learning Reporting Dynamics during Breaking News for Rumor Detection in Social Media," ARXIV, Oct. 2016.

[13] A. Zubiaga, M. Liakata, and R. Procter, "Exploiting Context for Rumor Detection in Social Media," Lecture Notes in Computer Science Social Informatics, pp. 109–123, Sep. 2017.

[14] M. Takayasu, K. Sato, Y. Sano, K. Yamada, W. Miura, and H. Takayasu, "Rumor Diffusion and Convergence during the 3.11 Earthquake: A Twitter Case Study," Plos One, vol. 10, no. 4, Apr. 2015.

[15] Y. Okada, K. Ikeda, K. Shinoda, F. Toriumi, T. Sakaki, K. Kazama, M. Numao, I. Noda, and S. Kurihara, "SIR-Extended Information Diffusion Model of False Rumor and its Prevention Strategy for Twitter," Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 18, no. 4, pp. 598–607, Jul. 2014.

[16] D. Malkhi, Y. Mansour, and M. K. Reiter, "Diffusion without false rumors: on propagating updates in a Byzantine environment," Theoretical Computer Science, vol. 299, no. 1-3, pp. 289–306, Apr. 2003.

[17] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. W. S. Hoi, and A. Zubiaga, "SemEval-2017 Task 8: RumorEval: Determining rumor veracity and support for rumors," Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Apr. 2017.

[18] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, no. 1-3, pp. 489–501, Dec. 2006.

[19] G.-B. Huang, "An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels," Cognitive Computation, vol. 6, no. 3, pp. 376–390, Apr. 2014.

[20] S. Yahia, S. Said, O. Jemai, M. Zaied, and C. B. Amar, "Comparison between extreme learning machine and wavelet neural networks in data classification," Ninth International Conference on Machine Vision (ICMV 2016), Mar. 2017.

[21] G. Giasemidis, C. Singleton, I. Agrafiotis, J. R. C. Nurse, A. Pilgrim, C. Willis, and D. V. Greetham, "Determining the Veracity of Rumors on Twitter," Lecture Notes in Computer Science Social Informatics, pp. 185–205, Oct. 2016.

[22] Y. Qin, D. Wurzer, V. Lavrenko, and C. Tang, "Spotting Rumors via Novelty Detection," ARXIV, Nov. 2016.

[23] R. Sicilia, S. L. Giudice, Y. Pei, M. Pechenizkiy, and P. Soda, "Health-related rumor detection on Twitter," 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Nov. 2017.

[24] R. Dayani, N. Chhabra, T. Kadian, and R. Kaushal, “Rumor detection in twitter: An analysis in retrospect,” 2015 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), Dec. 2015.

[25] S. Kumar, R. West, and J. Leskovec, “Disinformation on the Web,” Proceedings of the 25th International Conference on World Wide Web - WWW 16, Apr. 2016.

[26] B. Galitsky, “Detecting Rumor and Disinformation by Web Mining,” AAAI Spring Symposium Series, Mar. 2015.

[27] Z. Lao, Z. Zhou, and J. Huang, “Incremental Extreme Learning Machine via Fast Random Search Method,” Neural Information Processing Lecture Notes in Computer Science, pp. 82–90, Nov. 2017.

[28] J. Tang, C. Deng, and G.-B. Huang, “Extreme Learning Machine for Multilayer Perceptron,” IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 4, pp. 809–821, Apr. 2016.

[29] H. Song, A. K. Qin, and F. D. Salim, “Multi-resolution Selective Ensemble Extreme Learning Machine for Electricity Consumption Prediction,” Neural Information Processing Lecture Notes in Computer Science, pp. 600–609, Nov. 2017.

[30] D. Serre, Matrices: theory and applications. New York: Springer, 2011.

[31] C. R. Rao and S. K. Mitra, Generalized inverse of matrices and its applications. New York: Wiley, 1971.

[32] J. Shawe-Taylor and N. Cristianini, Kernel methods for pattern analysis. Cambridge, UK: Cambridge University Press, 2004.