Accredited by National Journal Accreditation (ARJUNA) Managed by
Ministry of Research, Technology, and Higher Education, Republic Indonesia with Second Grade (Peringkat 2, Sinta 2)
since year 2017 to 2021 according to the decree No. 10/E/KPT/2019.

Published online on the journal's webpage: **http://jurnal.iaii.or.id**

RESTI JOURNAL
(System Engineering and Information Technology)
Vol. 4 No. 5 (2020) 775 - 781 ISSN Electronic Media: 2580-0760

Implementation of Rumor Detection on Twitter Using J48 Algorithm

Yoan Maria Vianny[1], Erwin Budi Setiawan[2*]
[1,2]Informatics, School of Computing, Telkom University
[1]yoanmariav@student.telkomuniversity.ac.id, [2]erwinbudisetiawan@telkomuniversity.ac.id
[*]corresponding author

Abstract

The existence of rumors on Twitter has caused a lot of unrest among Indonesians. Unrecognized validity confuses users for that information. In this study, an Indonesian rumor detection system is built by using J48 Algorithm in collaboration with Term Frequency Inverse Document Frequency (TF-IDF) weighting method. Dataset contains 47.449 tweets that have been manually labeled. This study offers new features, namely the number of emoticons in display name, the number of digits in display name, and the number of digits in username. These three new features are used to maximize information about information sources. The highest accuracy is obtained by 75.76% using 90% training data and 1.000 TF-IDF features in 1-gram to 3-gram combinations.

Keywords: *Twitter, Rumor, Pre-processing, J48, TF-IDF*

1. Introduction

Social networks allow people to stay connected by exchanging information. Twitter has become one of the most popular social networks in Indonesia. The number of Twitter users in Indonesia ranks fifth in the world at around 19.5 million people [1]. Users can send text and multimedia which are known as tweet. This microblogging service also has an ongoing trending topic feature so users can find out the latest trends. But not all tweets spread on Twitter contain facts. There are much information where the truths are not known yet shared with fellow users known as rumor [2].

In emergencies such as natural disasters, Twitter is widely used as a communication and information media because of the rapid spread. In 2011, earthquake that caused Tsunami in Japan affected telephone networks in several area but the Internet could be accessed normally [3]. Twitter was used to share the latest information since communication services were limited [3]. The presence of rumors during an emergency such as a natural disaster or an economic crisis can be very dangerous to social security [4]. Flexibility of exchanging information on Twitter provides probabilities for rumor spreader to launch their action that can cause chaos [5]. They also do not feel hesitate to spread fear and hate speech [6].

Rumor is defined as a statement that has not been verified by official sources and spread by users on social networks [7]. The increasing number of users in some big platforms such as Twitter and Sina Weibo has made rumor becomes more serious social problem [8]. Rumor is formed from users who interact each other through the exchange of opinions with other users. In fact, people's thoughts differ from one another and many unfounded assumptions circulate in the community. This causes confusion and doubts the truth of information.

The phenomenon of political buzzers and fake accounts on social networks, especially Twitter becomes a concern. Often these elements massively spread statements that have not been proven to be true and dominate the trending topics. There are two types of buzzers namely the computer bot buzzer and the paid buzzer from fanatical supporters of a political group [9]. Many ways are done by these buzzers so that information that does not yet have proof seems trusted and valid.

These problems become background of many studies in the computer science with rumor as object. Several studies related to rumors have been carried out not only rumors in English, but also other languages that have non-alphabet script. One of them is the Chinese rumors detector on Sina Weibo based on feature selection and SDSMOTE [10]. Diffusion of rumors also has potential to disrupt stability in society and endanger one's life [11].

Accepted by editor: 19-06-2020 | Final Revision: 26-08-2020 | Online Publication: 30-10-2020

This concern becomes basis for the construction of Indonesian rumor detection to find out the content in a tweet containing rumors or non-rumors.

Rumor detection is an activity to determine whether an information has been verified or not yet distributed on social networks [5]. The uniqueness of Twitter users in Indonesia is the variety of word writing ranging from abbreviations, non-standard words, and combinations with foreign languages. This becomes a challenge especially in normalizing the writing into standard words which is intended to maximize the results of pre-processing. The weighting process with TF-IDF and classification using J48 algorithm are then performed.

J48 is an implementation of C4.5 algorithm that builds a decision tree and as a development of the ID3 algorithm [12]. This algorithm is also able to process discrete and continuous data also handle if there are attributes that contain missing values [12]. TF-IDF weighting method is involved in this study to determine the effect of words contained in the tweet on the classification process beside of using basic features such as mention, number of followings, and number of followers.

This study offers new features which are number of digits in display name, number of emoticons in display name, and number of digits in username. These additional features are useful for knowing the characteristics of users who spread rumor in more details. The purpose of this study is to implement methods that have been described above and find out performance of system in detecting the presence of Indonesian rumor.

2. Research Method

Data collection (crawling) is done using Twitter crawler developed by Jaka Sembodo et al. to obtain features from the Twitter index [13]. Crawling process is done based on keywords that can take a maximum of 100 tweets per search [13]. Topics taken are based on trending that occurred in Indonesia. Data collection period starts from October 2019 during the presidential inauguration period until February 2020 when COVID-19 is being heavily discussed. Data that have been collected are labeled as rumor or non-rumor manually by author. After labeling, tweet features are processed in pre-processing to clear unnecessary attribute to make classification process is not interrupted. Pre-processing sequence in this study consist of case folding, word normalization, stop words removal, and stemming.

In case folding, tweet is converted to lowercase and removes mention, number, URL, and retweet for uniformity of data. Abbreviations, contemporary terms, and foreign languages are normalized into standard Indonesian words by matching each word in the tweet with the normalized dictionary that has been made.

Words that have processed through normalization but have no effect on the classification are deleted through stop word removal process. Furthermore, stemming is done to eliminate prefixes or suffixes from a word into a basic word. Indonesian language stemming is processed using Python Sastrawi library. Examples of each pre-processing stages are showed in Table 1.

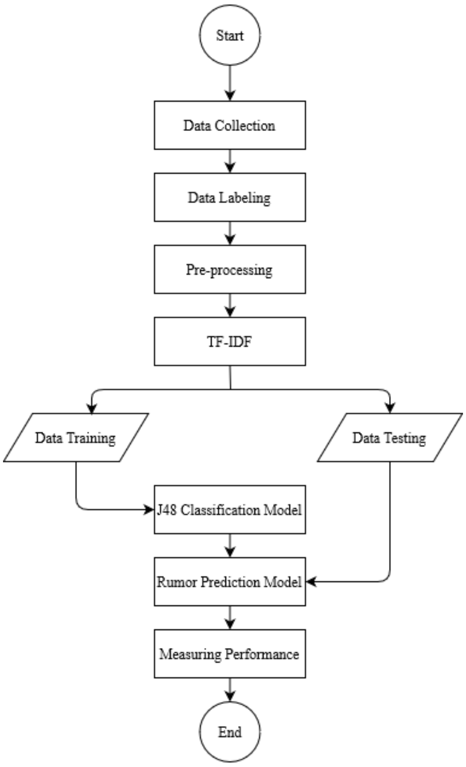


Figure 1. Research Flow

Table 1. Pre-processing Example

Step	Before	After
Case Folding	Swear... gak tahu apa ada acara besar kenegaraan hari ini? Kok ada hastag #MatikantvTgl20	swear gak tahu apa ada acara besar kenegaraan hari ini kok ada hastag
Word Normalization	swear gak tahu apa ada acara besar kenegaraan hari ini kok ada hastag	sumpah tidak tahu apa ada acara besar kenegaraan hari ini kok ada tagar
Stop Word Removal	sumpah tidak tahu apa ada acara besar kenegaraan hari ini kok ada tagar	sumpah acara kenegaraan tagar
Stemming	sumpah acara kenegaraan tagar	sumpah acara negara tagar

Remaining words in the final pre-processing are calculated by frequency with N-grams. N-gram model is used for predicting the next word [14]. N-gram tokenizes from a string into several forms according to number of words that want to be cut off. This study uses three main forms of N-gram namely unigram (1-gram), bigram (2-gram), and trigram (3-gram) as well as a combination of these three. Every word of N-gram results, besides knowing the word frequency is also weighted with TF-

IDF method which converts pre-processing results into numerical form so that they can be classified.

TF-IDF is an abbreviation of Term Frequency – Inverse Document Frequency that is a weighting method to calculate word value in a document through inverse proportion from word presence [15]. TF-IDF is calculated as follow:

$$W_{i,j} = tf_{i,j} * \log(N/df_i) \quad (1)$$

where Term Frequency (TF) is the frequency of word i in the document j which is represented by $tf_{i,j}$. Inverse Document Frequency (IDF) is represented by $\log(N/df_i)$ where N is total documents and df_i is total documents that contain i.

TF-IDF result is saved in .csv format and processed by J48 algorithm. J48 is a decision tree algorithm from C4.5 implementation [12]. The equation of J48 is described by [16] as follows:

$$Entropy = \sum_{i=1}^{n} -p_i * \log_2 p_i \quad (2)$$

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{S} * S_i \quad (3)$$

Notation of p_i stands for probability of class i in the range of values from 0 to 1. Data set is symbolized as S and attribute as A. In gain, S is entropy value (2) and S_i is entropy for data sample which is worth i. Number of data samples for i is stated as $|S_i|$. Partition ends when all tuples in node N get same classes or no attributes cannot be partitioned.

The classification process is run with the python-weka-wrapper3 library developed by the University of Waikato, New Zealand. Installation is almost the same as Weka Explorer which requires Java Development Kit. The difference is there must be a Javabridge installation and setup environment. Because python-weka-wrapper3 can only work in a separate environment so for using data frames from Pandas also the library of scikit-learn cannot be applied here.

Measuring performance becomes an important stage for evaluating results of classification. This activity is very useful to find out deficiencies or things that can be fixed or developed to improve quality of the system. Output of classification are a decision tree and a summary produced by Weka. Summary consists of accuracy (correctly classified instances), kappa statistics, mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), and root relative squared error (RRSE).

Table 2. Example of Suspected Rumor Spreaders

Display Name	Username	Tweet
b4n9y05	b4n9y05	"Saya tadi lewat depan gedung DPR, kok banyak polisi ya, pada hal gak ada demo. Jalan pada sepi, ada apa ya \n#MatikantvTgl20"
ɪᴅMT🔝🆔ᴘꜱ	MT_Reborn	"Aneh memang,dari Wuhan ke Batam baru ke Natuna lagi.\nKata Panglima TNI,Fasilitas Militer utk kesehatan sdh lengkap di Natuna.\n\ud83d\ude44 https:\/\/t.co\/KPQa9

In addition to basic features usage obtained through crawling, this study also develops new features to support identification of users who spread rumors. These features are number of digits in display name, number of emoticons in display name, and number of digits in username.

The background to using these features come from the discovery of behavior of rumors spreading users using pseudonyms combined with numbers or emoticons. A genuine identity that is not exposed to the public makes this user can tweet freely and anonymously. Table 2 shows some examples of accounts that allegedly spread rumors and tweets made. First example uses numbers that have similar shapes with letters for example the letter 'g' with the number 9 and the letter 's' with the number 5. In the second example, emoticons are used along with pseudonyms in foreign languages. Both examples cover up their true identities so they can freely spread rumors on Twitter without getting pressure from any party.

3. Result and Discussion

3.1. Dataset Test Result

The dataset used contain of 47.449 tweets that have been manually labeled by author. Before entering the classification, data are splitted for training and testing process. Scenario of data splitting is divided into 3 by using training data of 90%, 80%, and 50%. In addition to data splitting, the test is run based on the number of features along with composition of N-gram to compare classification performance. Explored number of TF-IDF features are 50, 200, 500, and 1.000. Composition of N-gram that is used for classification: 1-gram, 2-gram, 3-gram, combination of 1-gram and 2-gram, combination of 1-grams up to 3-grams, and combination of 2-grams and 3-grams. Results of N-gram and TF-IDF are saved in .csv format to be processed by python-weka-wrapper3. The highest accuracy for 90% training data is achieved by a combination of 1-gram up to 3-gram using 1.000 TF-IDF features with accuracy of 75.76%. The test results are shown in Table 3 and 4.

Table 3. Calibration of 90% Training Data

N-gram Composition	Accuracy	Kappa Statistics
1-gram	75.36%	0.4914
2-gram	74.25%	0.4593
3-gram	72.96%	0.4301
1-gram, 2-gram	75.53%	0.4938
1-gram, 2-gram, 3-gram	75.76%	0.497
2-gram, 3-gram	74.12%	0.4551

Distribution of 80% training data obtained the highest accuracy of 75.57% using 500 TF-IDF features. Just like testing 90% training data, combination of 1-gram up to 3-gram again achieves the highest accuracy compared to other N-gram compositions. The test results are shown in Table 5 and 6.

Table 4. Error Rate of 90% Training Data

N-gram Composition	MAE	RMSE	RAE	RRSE
1-gram	0.2921	0.4481	59.97%	90.69%
2-gram	0.316	0.4437	64.87%	89.79%
3-gram	0.3267	0.4502	67.07%	91.10%
1-gram, 2-gram	0.3297	0.452	67.80%	91.60%
1-gram, 2-gram, 3-gram	0.2894	0.4426	59.42%	89.57%
2-gram, 3-gram	0.3146	0.4475	64.60%	90.56%

Table 5. Calibration of 80% Training Data

N-gram Composition	Accuracy	Kappa Statistics
1-gram	74.67%	0.4746
2-gram	73.12%	0.4328
3-gram	72.62%	0.4176
1-gram, 2-gram	74.95%	0.478
1-gram, 2-gram, 3-gram	75.57%	0.4917
2-gram, 3-gram	73.16%	0.4322

Table 6. Error Rate of 80% Training Data

N-gram Composition	MAE	RMSE	RAE	RRSE
1-gram	0.2993	0.4507	61.54%	91.35%
2-gram	0.3207	0.4529	65.94%	91.78%
3-gram	0.3332	0.453	68.50%	91.81%
1-gram, 2-gram	0.2937	0.4487	60.39%	90.94%
1-gram, 2-gram, 3-gram	0.2931	0.4423	60.25%	89.64%
2-gram, 3-gram	0.3228	0.4521	66.36%	91.62%

Distribution of 50% training data gets the highest accuracy that is 73.32% using 1-gram composition and 500 TF-IDF features. The test results are shown in Table 7 and 8.

Table 7. Calibration of 50% Training Data

N-gram Composition	Accuracy	Kappa Statistics
1-gram	73.32%	0.4441
2-gram	72.23%	0.4158
3-gram	70.84%	0.3887
1-gram, 2-gram	72.79%	0.4339
1-gram, 2-gram, 3-gram	72.83%	0.4381
2-gram, 3-gram	71.92%	0.4075

Table 8. Error Rate of 50% Training Data

N-gram Composition	MAE	RMSE	RAE	RRSE
1-gram	0.3216	0.4571	66.16%	92.60%
2-gram	0.3352	0.4581	68.97%	92.81%
3-gram	0.3432	0.4698	70.60%	95.17%
1-gram, 2-gram	0.3195	0.4631	65.73%	93.82%
1-gram, 2-gram, 3-gram	0.3157	0.4652	64.94%	94.24%
2-gram, 3-gram	0.3362	0.4575	69.17%	92.69%

Each dataset test produces a decision tree. An example of a decision tree is shown in Figure 2.

Decision tree is sometimes difficult to understand so extraction rules are used to simplify the reading of decision trees. The rules are shown in Table 9.

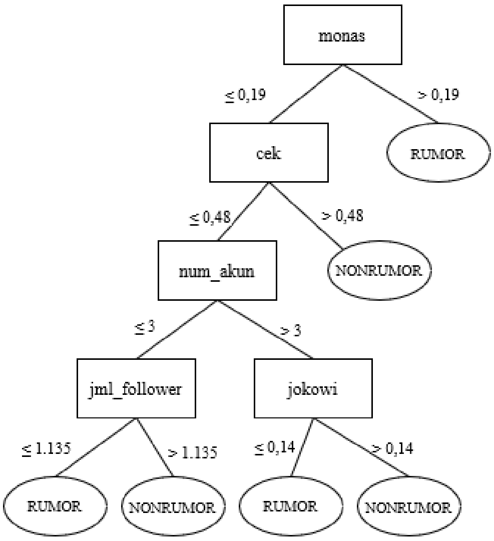


Figure 2. Decision Tree Example

Table 9. Rule Extraction from Decision Tree

Rule	
1	IF (monas ≤ 0,19) AND (cek ≤ 0,48) AND (num_akun ≤ 3) AND (jml_follower ≤ 1.135) THEN label = RUMOR
2	IF (monas ≤ 0,19) AND (cek ≤ 0,48) AND (num_akun ≤ 3) AND (jml_follower > 1.135) THEN label = NONRUMOR
3	IF (monas ≤ 0,19) AND (cek ≤ 0,48) AND (num_akun > 3) AND (jokowi ≤ 0,14) THEN label = RUMOR
4	IF (monas ≤ 0,19) AND (cek ≤ 0,48) AND (num_akun > 3) AND (jokowi > 0,14) THEN label = NONRUMOR
5	IF (monas ≤ 0,19) AND (cek > 0.48) THEN label = NONRUMOR
6	IF (monas > 0,19) THEN label = RUMOR

3.2 Analysis of Testing Results

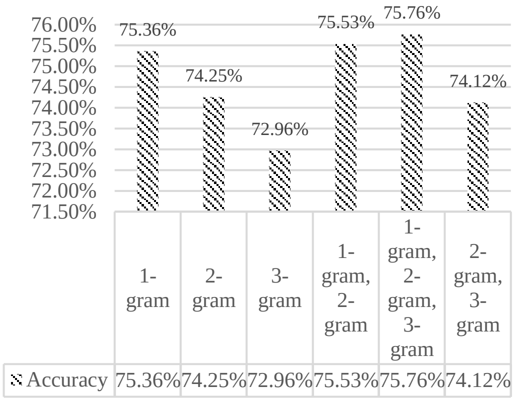


Figure 3. Accuracy of 90% Training Data and 1000 features

Combination of 1-gram up to 3-gram achieves the best performance for 90% and 80% training data splitting scenarios with the lowest error rate compared to other compositions. Whereas the best performance of 50% of training data is achieved by 1 gram. Similarity of these three scenarios is the role of 1-gram which helps to improve accuracy. It can be seen in figure 3 that application of a single 1-gram has the highest accuracy that is 75.36% rather than 2-gram which is 74.25% and 3-gram which achieves 72.96%. But when only 1-gram is applied without a combination even though the accuracy is high, error rate is still higher than combination of 1-gram up to 3-gram in Table 8. Thing that should be remembered is that quality of the system is not only seen from accuracy but also other components such as kappa statistics and error levels must also be considered.

In more detail, the results of each data distribution that has the highest accuracy also has the highest kappa statistics. The results can be seen in Figure 4.

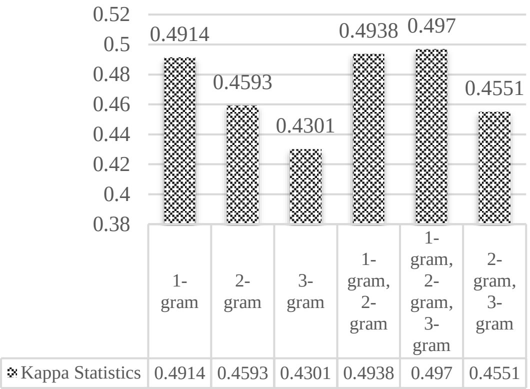


Figure 4. Kappa Statistics of 90% Training Data Usage

Higher kappa statistics indicates higher level of agreement between actual classes and prediction results. It can be said if the tweet is labeled a rumor, then the probability of prediction the tweet as a rumor is also higher, also the prediction error rate is lower.

The number of features selection has an influence on the final performance. The 90% training data scenario achieves the highest accuracy uses 1.000 TF-IDF features. In 80% and 50% data training scenarios, the TF-IDF feature of 500 obtain the highest yield compared to the feature of 1.000. This means that the greater training data used, the more N-gram filtration results are needed to be learned by classifier.

Table 10. N-gram Results

1-gram	2-gram	3-gram
“ragu”	“virus corona”	“polemik revitalisasi tim”
“isu”	“revitalisasi monas”	“warga natuna tolak”
“takut”	“ekspor ganja”	“sebar virus corona”
“hancur”	“isu reshuffle”	“hukum mati koruptor”
“panik”	“alih isu”	“daun monas gersang”

Table 10 displays some features of N-gram filtration. These words often appear in the dataset and have been weighted by TF-IDF. The more frequency of words, indication of a rumor is also high because many people talk about the word. Opinions and assumptions that are made by users become initial formation of a rumor. Tweet labeling has big part in determining the accuracy of prediction class as well as the training process so it must be done as well as possible to achieve the best performance.

3.3. Effects of Adding New Features

Number of emoticons in display name has the biggest contribution in improving the system performance shows in Figure 5. Before there was a new feature, the accuracy obtained was only 74.42%. But after an experiment using one new feature, accuracy has increased. Single feature usage of the number of emoticons in display name increases accuracy to 74.98%. Single feature usage of the number of digits in display name achieves only 0.04% adrift resulting in 74.94%. Whereas the single feature usage of the number of digits in username increases, unlike the other two features, which is only 74.67% but still helps when all three features are used together. The application of three features using 90% training data with combination of 1-gram up to 3-gram and 1.000 TF-IDF features succeeds in increasing accuracy to 75.76%.

Table 11 shows that many accounts that use emoticons and numbers either display names or usernames are involved in spreading rumors on Twitter. Two examples of accounts labeled with rumors do not include real names but contemporary also foreign terms and none are verified users. While accounts labeled as non-rumors are accounts from the Indonesian mainstream media that have also been verified by Twitter.

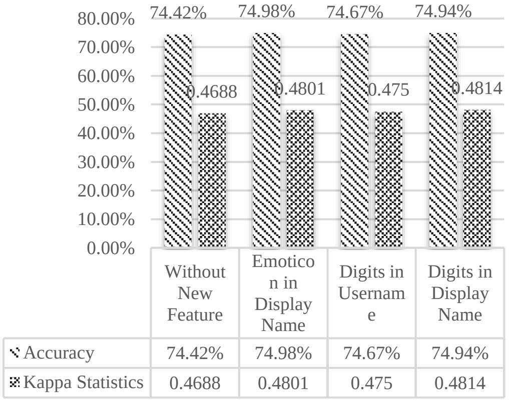
Table 11. Label of Users

Label	Display Name	Username
RUMOR	CoolCat🔊	BonekaDongo
	AQ Sopo auto follback♡💞ID💞♡ID♡💞ID💞♡	AQSopo4
NONRUMOR	Liputan6.com	Liputan6dotcom
	KOMPAS TV	KompasTV

Verified mainstream media accounts spread official news through tweets in standard language according to journalism ethics and standards. Writing tweet is done neutrally and carefully so as not to cause disinformation. Then the display name and username are made simple and in accordance with the name of the mainstream media. Its function is to make it easier for users to find the account, especially when they want to know truth of a news. No mainstream media are found that included emoticons on the display name. But there are some who use numbers because they adjust to the name of their mainstream media.

The existence of disinformation is an opportunity for individuals to make the latest rumors. Rumor spreader accounts often distort facts of official news for certain interests such as imposing someone, creating a chaos, or making an image of a public figure look good. Hidden identity is an advantage when launching actions on Twitter. There is no fear and pressure from third party that contradicts opinions and behavior. Suppose the account is reported to Twitter to be deactivated, this person can create a new account and act again. Not infrequently this person creates a backup account where in addition to anticipating account deactivation, it also helps in intensifying the spread of rumors.

Discovery of a rumor spreader account that covers identity using emoticons or numbers can enrich user information. This can be interpreted that digging information from a main feature into several sub features can help to identify rumors for classification process. It is proven that the kappa statistics value increased by 0.0282 after implementing new features even though the accuracy obtained did not touch 80%.



	Without New Feature	Emoticon in Display Name	Digits in Username	Digits in Display Name
▧ Accuracy	74.42%	74.98%	74.67%	74.94%
▩ Kappa Statistics	0.4688	0.4801	0.475	0.4814

Figure 5. Comparison of New Feature

There are still many non-standard words, contemporary terms, and foreign terms that have not been accommodated in the normalization dictionary. For example, the word 'keren' is written in double letters or other writing methods such as 'keereeennn' or 'qereeenn'. This variety of writing makes the pre-processing results do not fully contain standard Indonesian. This is a serious challenge to get clean pre-processing results because pre-processing has a big influence on the weighting process to the process of measuring system performance.

Rumor spreaders have certain characteristics so that their identity remains confidential and free to act on Twitter. The performance results of this new feature indicate that it is necessary to analyze user habits on social networks so that the accuracy of the system in predicting rumors can be even better.

4. Conclusion

This study uses Indonesian-language rumors as the main attention to build a rumor detection system using the J48 algorithm. New features offered are number of emoticons in display name, number of digit in display names, and number of digits in username proved to be able to improve the performance of the rumor detection system and minimize the level of errors. Before using new features, accuracy obtained was only 74.42% but increased after implementing these three new features. The highest accuracy is obtained by 75.76% from the 90% training data that processes 1.000 TF-IDF features containing a combination of 1-gram, 2-gram, and 3-gram and involves 3 new features. Suggestions for further research are the development of normalization dictionaries and Indonesian stop words to improve the quality of pre-processing and accuracy of Indonesian rumor detection.

References

[1] E. B. Setiawan, D. H. Widyantoro, and K. Surendro, "Feature expansion using word embedding for tweet topic classification," *Proceeding 2016 10th Int. Conf. Telecommun. Syst. Serv. Appl. TSSA 2016 Spec. Issue Radar Technol.*, 2017, doi: 10.1109/TSSA.2016.7871085.

[2] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on Sina Weibo by propagation structures," *Proc. - Int. Conf. Data Eng.*, vol. 2015-May, pp. 651–662, 2015, doi: 10.1109/ICDE.2015.7113322.

[3] S. Luna and M. Pennock, "Social media in emergency management advances, challenges and future directions," *9th Annu. IEEE Int. Syst. Conf. SysCon 2015 - Proc.*, pp. 792–797, 2015, doi: 10.1109/SYSCON.2015.7116847.

[4] Z. Wang, Y. Guo, J. Wang, Z. Li, and M. Tang, "Rumor Events Detection From Chinese Microblogs via Sentiments Enhancement," *IEEE Access*, vol. 7, pp. 103000–103018, 2019, doi: 10.1109/access.2019.2928044.

[5] S. A. Alkhodair, S. H. H. Ding, B. C. M. Fung, and J. Liu, "Detecting breaking news rumors of emerging topics in social media," *Inf. Process. Manag.*, vol. 57, no. 2, p. 102018, 2020, doi: 10.1016/j.ipm.2019.02.016.

[6] Q. Li, Q. Zhang, and L. Si, "Rumor detection by exploiting user credibility information, attention and multi-task learning," *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 1173–1179, 2020, doi: 10.18653/v1/p19-1113.

[7] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Inf. Sci. (Ny).*, vol. 497, pp. 38–55, 2019, doi: 10.1016/j.ins.2019.05.035.

[8] C. Song, C. Yang, H. Chen, C. Tu, Z. Liu, and M. Sun, "CED: Credible Early Detection of Social Media Rumors," *IEEE Trans. Knowl. Data Eng.*, vol. PP, no. 8, pp. 1–1, 2019, doi: 10.1109/tkde.2019.2961675.

[9] M. Ibrahim, O. Abdillah, A. F. Wicaksono, and M. Adriani, "Buzzer Detection and Sentiment Analysis for Predicting Presidential Election Results in a Twitter Nation," *Proc. - 15th IEEE Int. Conf. Data Min. Work. ICDMW 2015*, pp. 1348–1353, 2016, doi: 10.1109/ICDMW.2015.113.

[10] Y. Geng, J. Sui, and Q. Zhu, "Rumor detection of Sina Weibo based on SDSMOTE and feature selection," *2019 IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2019*, pp. 120–125, 2019, doi: 10.1109/ICCCBDA.2019.8725715.

[11] J. Li, H. Ji, D. Zhao, and Y. Feng, "Automatic Detection of Rumor on Social Network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9362, pp. 113–122, 2015, doi: 10.1007/978-3-319-25207-0.

[12] E. B. Setiawan, D. H. Widyantoro, and K. Surendro,

"Measuring information credibility in social media using combination of user profile and message content dimensions," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 3537–3549, 2020, doi: 10.11591/ijece.v10i4.pp3537-3549.

[13] J. Eka Sembodo, E. Budi Setiawan, and Z. Abdurahman Baizal, "Data Crawling Otomatis pada Twitter," no. August, pp. 11–16, 2016, doi: 10.21108/indosc.2016.111.

[14] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, 2016, doi: 10.1016/j.eswa.2016.03.028.

[15] C. P. Medina and M. R. R. Ramon, "Using TF-IDF to Determine Word Relevance in Document Queries Juan," *New Educ. Rev.*, vol. 42, no. 4, pp. 40–51, 2015, doi: 10.15804/tner.2015.42.4.03.

[16] S. Diwandari and N. A. Setiawan, "Perbandingan Algoritme J48 Dan Nbtree Untuk Klasifikasi Diagnosa Penyakit Pada Soybean," *Semin. Nas. Teknol. Inf. dan Komun.*, vol. 2015, no. Sentika, pp. 205–212, 2015.