

# Fake Reviews Detection using Supervised Machine Learning

Ahmed M. Elmogy<sup>1</sup>, Usman Tariq<sup>2</sup>, Atef Ibrahim<sup>4</sup>  
College of Computer Engineering and Sciences  
Prince Sattam Bin Abdulaziz University, KSA<sup>1,2,4</sup>  
Faculty of Eng., Tanta University,  
Egypt<sup>1</sup>

Ammar Mohammed<sup>3</sup>  
Department of Computer Science  
Misr International University, Egypt  
Faculty of Graduate Studies of Statistical Research  
Cairo University, Egypt

**Abstract**—With the continuous evolve of E-commerce systems, online reviews are mainly considered as a crucial factor for building and maintaining a good reputation. Moreover, they have an effective role in the decision making process for end users. Usually, a positive review for a target object attracts more customers and lead to high increase in sales. Nowadays, deceptive or fake reviews are deliberately written to build virtual reputation and attracting potential customers. Thus, identifying fake reviews is a vivid and ongoing research area. Identifying fake reviews depends not only on the key features of the reviews but also on the behaviors of the reviewers. This paper proposes a machine learning approach to identify fake reviews. In addition to the features extraction process of the reviews, this paper applies several features engineering to extract various behaviors of the reviewers. The paper compares the performance of several experiments done on a real Yelp dataset of restaurants reviews with and without features extracted from users behaviors. In both cases, we compare the performance of several classifiers; KNN, Naive Bayes (NB), SVM, Logistic Regression and Random forest. Also, different language models of n-gram in particular bi-gram and tri-gram are taken into considerations during the evaluations. The results reveal that KNN(K=7) outperforms the rest of classifiers in terms of f-score achieving best f-score 82.40%. The results show that the f-score has increased by 3.80% when taking the extracted reviewers behavioral features into consideration.

**Keywords**—Fake reviews detection; data mining; supervised machine learning; feature engineering

## I. INTRODUCTION

Nowadays, when customers want to draw a decision about services or products, reviews become the main source of their information. For example, when customers take the initiation to book a hotel, they read the reviews on the opinions of other customers on the hotel services. Depending on the feedback of the reviews, they decide to book room or not. If they came to a positive feedback from the reviews, they probably proceed to book the room. Thus, historical reviews became very credible sources of information to most people in several online services. Since, reviews are considered forms of sharing authentic feedback about positive or negative services, any attempt to manipulate those reviews by writing misleading or inauthentic content is considered as deceptive action and such reviews are labeled as fake [1]. Such case leads us to think what if not all the written reviews are honest or credible. What if some of these reviews are fake. Thus, detecting fake review has become and still in the state of active and required research

area [2].

Machine learning techniques can provide a big contribution to detect fake reviews of web contents. Generally, web mining techniques [3] find and extract useful information using several machine learning algorithms. One of the web mining tasks is content mining. A traditional example of content mining is opinion mining [4] which is concerned of finding the sentiment of text (positive or negative) by machine learning where a classifier is trained to analyze the features of the reviews together with the sentiments. Usually, fake reviews detection depends not only on the category of reviews but also on certain features that are not directly connected to the content. Building features of reviews normally involves text and natural language processing NLP. However, fake reviews may require building other features linked to the reviewer himself like for example review time/date or his writing styles. Thus the successful fake reviews detection lies on the construction of meaningful features extraction of the reviewers.

To this end, this paper applies several machine learning classifiers to identify fake reviews based on the content of the reviews as well as several extracted features from the reviewers. We apply the classifiers on real corpus of reviews taken from Yelp [5]. Besides the normal natural language processing on the corpus to extract and feed the features of the reviews to the classifiers, the paper also applies several features engineering on the corpus to extract various behaviors of the reviewers. The paper compares the impact of extracted features of the reviewers if they are taken into consideration within the classifiers. The papers compares the results in the absence and the presence of the extracted features in two different language models namely TF-IDF with bi-grams and TF-IDF with tri-grams. The results indicates that the engineered features increase the performance of fake reviews detection process.

The rest of this paper is organized as follows: Section II Summarizes the state of art in detecting fake reviews. Section III introduces a background about the machine learning techniques. Section IV presents the details of the proposed approach. Conclusions and future work are introduced in Section VI.

## II. RELATED WORK

The fake reviews detection problem has been tackled since 2007 [6]. Two main categories of features have been exploited

in the Fake reviews detection research; textual and behavioral features. Textual features refer to the verbal characteristic of review activity. In other words, textual features depend mainly on the content of the reviews. Behavioral features refer to the nonverbal characteristics of the reviews. They depend mainly on the behaviors of the reviewers such as writing style, emotional expressions, and the frequent times the reviewers write the reviews. Although tackling textual features is challenging and crucial, behavioral features are also very important and cannot be ignored as they have a high impact on the performance of the fake review detection process. Textual features have extensively been seen in several fake reviews detection research papers. In [7], the authors used supervised machine learning approaches for fake reviews detection. Five classifiers are used which are SVM, Naive-bayes, KNN, k-star and decision tree. Simulation experiments have been done on three versions of labeled movie reviews dataset [8] consisting of 1400, 2000, and 10662 movie reviews respectively. Also, in [9], the authors used Naive Bayes, Decision tree, SVM, Random forest and Maximum entropy classifiers in detecting fake reviews on the dataset that they have collected. The collected dataset is around 10,000 negative tweets related to Samsung products and their services. In [10], the authors used both SVM and Naive base classifiers. The authors worked on yield dataset which consists of 1600 reviews collected from 20 popular hotels in Chicago. In [11], the authors used the neural and discrete models with Average, CNN, RNN, GRNN, Average GRNN and Bi-directional Average GRNN deep learning classifiers to detect deceptive opinion spamming. They used dataset from [12] which contains truthful and deceptive reviews in three domains; namely hotels, restaurants and doctors. All the above research works have only considered the textual features without any effort towards the behavioral features.

Other articles have considered behavioral features in the fake reviews detection process. In [13], some behavioral features have been considered on Amazon reviews such as average rating, and ratio of the number of reviews that the reviewer wrote. In another work [14], the authors investigated the impact of both textual and behavioural features on the fake review detection process focusing on the restaurant and hotel domain. Also, In[15], an iterative computation framework plus (ICF++) is proposed integrating textual and behavioral features. They detected fake reviews based on measuring the honesty value of a review, the trustiness value of the reviewers and the reliability value of a product.

From the above discussion and to the best of our knowledge, no approaches have dived deeply in extracting features that reflect the reviewers' behaviors. These features will highly influence the effectiveness of the fake reviews detection process. In this paper a machine learning approach to identify fake reviews is presented. In addition to the features extraction process of the reviews, the presented approach applies several features engineering to extract various behaviors of the reviewers. Some new behavioral features are created. The created features are used as inputs to the proposed system besides the textual features for fake reviews detection task.

### III. BACKGROUND

Machine learning is one of the most important technological trends which lies behind many critical applications. The main power of machine learning is helping machines to automatically learn and improve themselves from previous experience [16]. There are several types of machine learning algorithms [17]; namely supervised, semi supervised and unsupervised machine learning. In the supervised approach, both input and output data are provided and the training data must be labeled and classified [18]. In the unsupervised learning approach, only the data is given without any classification or labels and the role of the approach is to find the best fit clustering or classification of the input data. Thus, in unsupervised learning, all data are unlabeled and the role of the approach is to label them. Finally, in the semi supervised approach, some data are labeled but the most are unlabeled. In this part, we introduce a summary of the supervised learning algorithms as they are the main focus of this paper.

Several classification algorithms are developed for supervised machine learning. The main objective of these algorithms is to find a proper model that disseminates the training data. For example, **Support Vector Machines (SVM)** is a discriminated classifier that basically separates the given data into classes by finding the best separable hyper-plane which categorizes the given training data [19]. Another Common supervised learning algorithm is **Naive Bayes (NB)**. The key idea of NB relies on Bayes theorem; the probability of event A to happen given the probability of event B which is formed as  $P(A|B) = P(B|A) * P(A) / P(B)$  [20]. NB calculates a set of probabilities by counting the frequency and the combined values in a given dataset. NB has been successfully applied in several application domains like text classification, spam filtering and recommendation systems.

**The K-Nearest Neighbors algorithm (or KNN)** [21] is one of the most simple yet powerful classification algorithms. KNN has been used mostly in statistical estimation and pattern recognition. The key idea behinds KNN is to classify instance query based on voting of a group of similar classified instances. The similarity is usually calculated using distance function [22]

**Decision-tree** [23] is another machine learning classifier that relies on building a tree that represents a decision of instances training data. The Algorithm starts to construct the tree iteratively based on best possible split among features. The selection process of the best features relies on a predefined functions like, entropy, information gain, gain ratio, or gini index. **Random Forest** [24] is a successful method that handles the overfitting problems that occur in the decision tree. The key essence of random forest is to construct a bag of trees from different samples of the dataset. Instead of constructing the tree from all features, Random forest generates small random number of features while constructing each tree in the forest. **Logistic regression** [25] is another simple supervised machine learning classifier. It relies on finding a hyperplane that classifies the data.

### IV. PROPOSED APPROACH

This section explains the details of the proposed approach shown in figure 1. The proposed approach consists of three basic phases in order to get the best model that will be used

for fake reviews detection. These phases are explained in the following:

### A. Data Preprocessing

The first step in the proposed approach is data preprocessing [26]; one of the essential steps in machine learning approaches. Data preprocessing is a critical activity as the world data is never appropriate to be used. A sequence of preprocessing steps have been used in this work to prepare the raw data of the Yelp dataset for computational activities. This can be summarized as follows:

1) *Tokenization*: Tokenization is one of the most common natural language processing techniques. It is a basic step before applying any other preprocessing techniques. The text is divided into individual words called tokens. For example, if we have a sentence (“wearing helmets is a must for pedal cyclists”), tokenization will divide it into the following tokens (“wearing”, “helmets”, “is”, “a”, “must”, “for”, “pedal”, “cyclists”) [27].

2) *Stop Words Cleaning*: Stop words [28] are the words which are used the most yet they hold no value. Common examples of the stop words are (an, a, the, this). In this paper, all data are cleaned from stop words before going forward in the fake reviews detection process.

3) *Lemmatization*: Lemmatization method is used to convert the plural format to a singular one. It is aiming to remove inflectional endings only and to return the base or dictionary form of the word. For example: converting the word (“plays”) to (“play”) [29].

Several approaches have been developed in the literature to extract features for fake reviews detection. Textual features is one popular approach [31]. It contains sentiment classification [32] which depends on getting the percent of positive and negative words in the review; e.g. “good”, “weak”. Also, the Cosine similarity is considered. The Cosine similarity is the cosine of the angle between two n-dimensional vectors in an n-dimensional space and the dot product of the two vectors divided by the product of the two vectors’ lengths (ormagnitudes)[33]. TF-IDF is another textual feature method that gets the frequency of both true and false (TF) and the inverse document (IDF). Each word has a respective TF and IDF score and the product of the TF and IDF scores of a term is called the TF-IDF weight of that term [34]. A confusion matrix is used to classify the reviews into four results; True Negative (TN): Real events are classified as real events, True Positive (TP): Fake events are classified as fake, False Positive (FP): Real events are classified as fake events, and False Negative (FN): Fake events are classified as real.

Second there are user personal profile and behavioral features. These features are the two ways used to identify spammers Whether by using time-stamp of user’s comment is frequent and unique than other normal users or if the user posts a redundant review and has no relation to domain of target.

In this paper, We apply TF-IDF to extract the features of the contents in two languages models; mainly bi-gram and tri-gram. In both language models, we apply also the extended dataset after extracting the features representing the users behaviors.

### C. Feature Engineering

Fake reviews are known to have other descriptive features [35] related to behaviors of the reviewers during writing their reviews. In this paper, we consider some of these feature and their impact on the performance of the fake reviews detection process. We consider caps-count, punct-count, and emojis behavioral features. *caps-count* represents the total capital character a reviewer use when writing the review, *punct-count* represents the total number of punctuation that found in each review, and *emojis* counts the total number of emojis in each review. Also, we have used statistical analysis on reviewers’ behaviours by applying “groupby” function, that gets the number of fake or real reviews by each reviewer that are written on a certain date and on each hotel. All these features are taken into consideration to see the effect of the users behaviors on the performance of the classifiers.

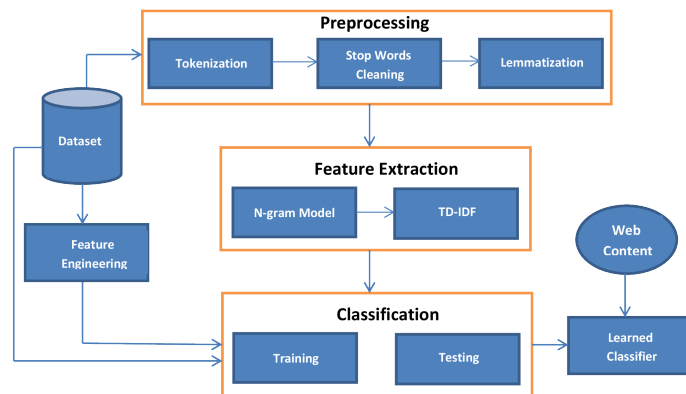


Fig. 1. The Proposed Framework.

### B. Feature Extraction

Feature extraction is a step which aims to increase the performance either for a pattern recognition or machine learning system. Feature extraction represents a reduction phase of the data to its important features which yields in feeding machine and deep learning models with more valuable data. It is mainly a procedure of removing the unneeded attributes from data that may actually reduce the accuracy of the model [30].

## V. EXPERIMENTAL RESULTS

We evaluated our proposed system on Yelp dataset [5]. This dataset includes 5853 reviews of 201 hotels in Chicago written by 38,063 reviewers. The reviews are classified into 4,709 review labeled as real and 1,144 reviews labeled as fake. Yelp has classified the reviews into genuine and fake. Each instance of the review in the dataset contains the review date, review ID, reviewer ID, product ID, review label and star rating. The statistics of dataset is summarized in Table I. The maximum review length in the data contains 875 word, the minimum review length contains 4 words, the average length

of all the reviews is 439.5 word, the total number of tokens of the data is 103052 word, and the number of unique words is 102739 word.

TABLE I. SUMMARY OF THE DATASET

Total number of reviews	5853 review
Number of fake reviews	1144 review
Number of real reviews	4709 review
Number of distinct words	102739 word
Total number of tokens	103052 token
The Maximum review length	875 word
The Minimum review length	4 word
The Average review length	439.5 word

In addition to the dataset and its statistics, we extracted other features representing the behaviors of reviewers during writing their reviews. These features include *caps-count* which represents the total capital character a reviewer use when writing the review, *punct-count* which represents the total number of punctuation that found in each review, and *emojis* which counts the total number of emojis in each review. We will take all these features into consideration to see the effect of the users behaviors on the performance of the classifiers.

In this part, we present the results for several experiments and their evaluation using five different machine learning classifiers. We first apply TF-IDF to extract the features of the contents in two languages models; mainly bi-gram and tri-gram. In both language models, we apply also the extended dataset after extracting the features representing the users behaviors mentioned in the last section. Since the dataset is unbalanced in terms of positive and negative labels, we take into consideration the precision and the recall, and hence and hence f1-score is considered as a performance measure in addition to accuracy. 70% of the dataset is used for training while 30% is used for testing. The classifiers are first evaluated in the absence of extracted features behaviors of users and then in the presence of the extracted behaviors. In each case, we compare the performance of classifiers in Bi-gram and Tri-gram language models.

Table II Summarizes the results of accuracy in the absence of extracted features behaviors of users in the two language models. The average accuracy for each classifier of the two language models is shown. It is found that the logistic regression classifier gives the highest accuracy of 87.87% in Bi-gram model. SVM and Random forest classifiers have relatively close accuracy to logistic regression. In Tri-gram model, KNN and Logistic regression are the best with accuracy of 87.87%. SVM and Random forest have relatively close accuracy with score of 87.82%. In order to evaluate the overall performance, we take into consideration the average accuracy of each classifier in both language models. It is found that the highest average accuracy is achieved in logistic regression with 87.87%. The summary of the results are shown in Fig. 2.

On the other hand, Table III summarizes the accuracy of the classifiers in the presence of the extracted features behaviors of the users in the two language models. The results reveal that the classifiers that give the highest accuracy in Bi-gram is SVM with score of 86.9%. Logistic regression and Random forest have relatively close accuracy with score of 86.89% and 86.85%, respectively. While in Tri-gram model, both SVM, and logistic regression give the best accuracy with score of

TABLE II. ACCURACY OF BI-GRAM AND TRI-GRAM IN THE ABSENCE OF EXTRACTED FEATURES BEHAVIORS

Classification Algorithm	Accuracy% Bigram	Accuracy% Trigram	Average Accuracy
Logistic Regression	<b>87.87%</b>	<b>87.87%</b>	<b>87.87%</b>
Naive bayes	86.76%	87.30%	87.03%
KNN (K=7)	86.34%	<b>87.87%</b>	87.82%
SVM	87.82%	87.82%	87.82%
Random Forest	87.82%	87.82%	87.82%

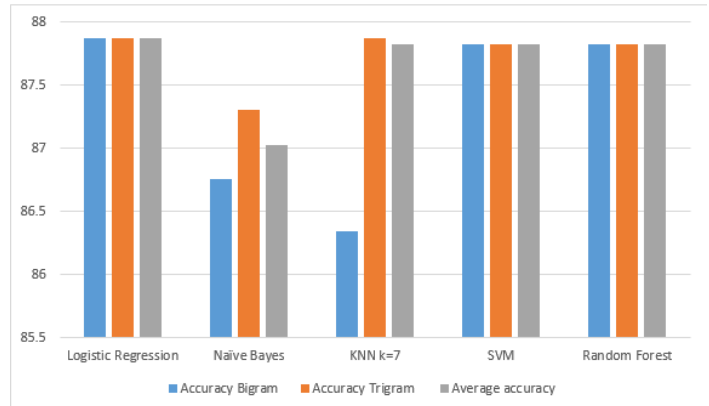


Fig. 2. Accuracy, and Average Accuracy in Absence of Extracted Behavioral Features.

86.9%. The Random forest gives a close score of 86.8%. The summary of the results are illustrated in Fig. 3. Also, it is found that the highest average accuracy is obtained with SVM classifier with score of 86.9%.

TABLE III. ACCURACY OF BI-GRAM AND TRI-GRAM IN THE PRESENCE OF EXTRACTED FEATURES BEHAVIORS

Classification Algorithm	Accuracy% Bigram	Accuracy% Trigram	Average Accuracy
Logistic Regression	86.89%	<b>86.9%</b>	86.89%
Naive bayes	85.82%	86.34%	86.08%
KNN (K=7)	86.56%	85.9%	86.23%
SVM	<b>86.9%</b>	<b>86.9%</b>	<b>86.9%</b>
Random Forest	86.85%	86.8%	86.82%

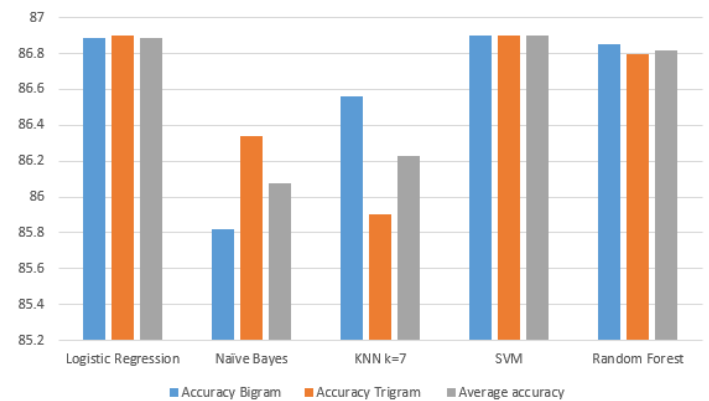


Fig. 3. The Accuracy, and the Average Accuracy after Applying Feature Engineering.

Additionally, precision, recall and f1-score are taken into consideration as evaluation metrics. Actually, they are key

indicators when the data is unbalanced. Similar to the previous, Table IV represents the recall, precision, and hence f1-score in the absence of the extracted features behaviors of the users in the two language models. For the trade-off between recall and precision, f1-score is taken into account as the evaluation criterion of each classifier. In Bi-gram, KNN(k=7) outperforms all other classifiers with f1-score value of 82.40%. Whereas, in Tri-gram, both logistic regression and KNN(K=7) outperform other classifiers with f1-score value of 82.20%. To evaluate the overall performance of the classifiers in both language models, the average f1-score is calculated. It is found that, KNN outperforms the overall classifiers with average f1-score of 82.30%. Fig. 4 depicts the the overall performance of all classifiers.

TABLE IV. RECALL, PRECISION, AND F1-SCORE IN ABSENCE OF EXTRACTED BEHAVIORAL FEATURES

	Bi-gram			Tri-gram			Avg F-score
	Recall	Precision	F-score	Recall	Precision	F-score	
Logistic Regression	<b>87.87%</b>	77.22%	82.20%	<b>87.87%</b>	77.20%	<b>82.20%</b>	82.20%
Naive Bayes	86.79%	78.23%	81.86%	87.30%	<b>78.97%</b>	82.12%	81.99%
KNN(K=7)	86.34%	80.20%	<b>82.40%</b>	<b>87.87%</b>	77.22%	<b>82.20%</b>	<b>82.30%</b>
SVM	87.82%	77.21%	82.17%	87.82%	77.21%	82.17%	82.17%
Random Forest	87.82%	<b>81.29%</b>	82.28%	87.82%	77.21%	82.17%	82.22%

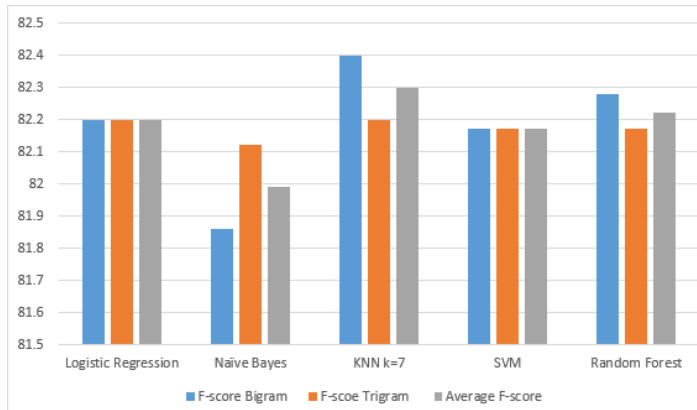


Fig. 4. f-score, and Average f-score in Absence of Extracted Behavioral Features.

Similarly, Table V summarizes the recall, precision, and f1-score in the presence of the extracted features behaviors of the users in the two language models. It is found that, the highest f1-score value is achieved by Logistic regression with f1-score value of 82% in case of Bi-gram. While the highest f1-score value in Tri-gram is achieved in KNN with f1-score value of 86.20%. Fig. 5 illustrates the performance of all classifiers. The KNN classifier outperforms all classifiers in terms of the overall average f1-score with value of 83.73%.

The results reveal that KNN(K=7) outperforms the rest of classifiers in terms of f-score with the best achieving f-score 82.40%. The result is raised by 3.80% when taking the extracted features into consideration giving best f-score value of 86.20%.

## VI. CONCLUSION

In this paper, we showed the importance of reviews and how they affect almost every thing related to web based data.

TABLE V. RECALL, PRECISION, AND F1-SCORE IN PRESENCE OF EXTRACTED BEHAVIORAL FEATURES

	Bi-gram			Tri-gram			Avg F-score
	Recall	Precision	F-score	Recall	Precision	F-score	
Logistic Regression	<b>86.90%</b>	75.53%	<b>82%</b>	86.90%	75.53%	80.82%	81.41%
Naive Bayes	85.82%	76%	80.38%	86.34%	76.59%	80.64%	80.51%
KNN(K=7)	86.56%	<b>80%</b>	81.26%	85.30%	<b>78.50%</b>	<b>86.20%</b>	<b>83.73%</b>
SVM	<b>86.90%</b>	75.50%	80.82%	84.90%	75.53%	81.82%	81.32%
Random Forest	86.85%	75.50%	80.79%	<b>87.90%</b>	74.53%	81.90%	81.34%

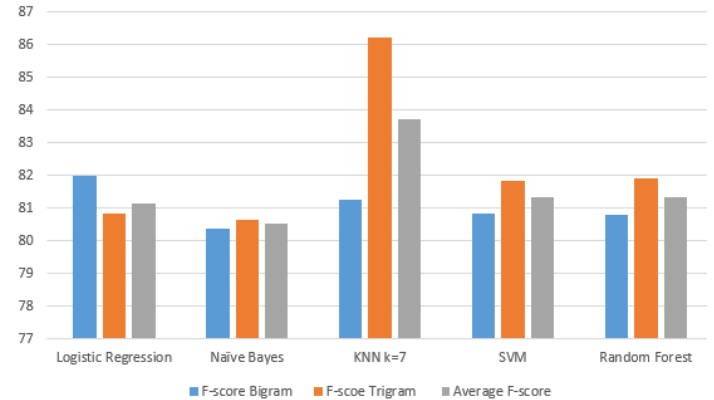


Fig. 5. f-score, and Average f-score in Presence of Extracted Behavioral Features.

It is obvious that reviews play a crucial role in people’s decision. Thus, fake reviews detection is a vivid and ongoing research area. In this paper, a machine learning fake reviews detection approach is presented. In the proposed approach, both the features of the reviews and the behavioral features of the reviewers are considered. The Yelp dataset is used to evaluate the proposed approach. Different classifiers are implemented in the developed approach. The Bi-gram and Tri-gram language models are used and compared in the developed approach. The results reveal that KNN(with K=7) classifier outperforms the rest of classifiers in the fake reviews detection process. Also, the results show that considering the behavioral features of the reviewers increase the f-score by 3.80%. Not all reviewers behavioral features have been taken into consideration in the current work. Future work may consider including other behavioral features such as features that depend on the frequent times the reviewers do the reviews, the time reviewers take to complete reviews, and how frequent they are submitting positive or negative reviews. It is highly expected that considering more behavioral features will enhance the performance of the presented fake reviews detection approach.

## ACKNOWLEDGMENTS

The authors would like to thank the Deanship of Scientific Research in Prince Sattam Bin Abdulaziz University, KSA for his support during the stages of this research.

## REFERENCES

- [1] R. Barbado, O. Araque, and C. A. Iglesias, “A framework for fake review detection in online consumer electronics retailers,” *Information Processing & Management*, vol. 56, no. 4, pp. 1234 – 1244, 2019.

- [2] S. Tadelis, "The economics of reputation and feedback systems in e-commerce marketplaces," *IEEE Internet Computing*, vol. 20, no. 1, pp. 12–19, 2016.
- [3] M. J. H. Mughal, "Data mining: Web data mining techniques, tools and algorithms: An overview," *Information Retrieval*, vol. 9, no. 6, 2018.
- [4] C. C. Aggarwal, "Opinion mining and sentiment analysis," in *Machine Learning for Text*. Springer, 2018, pp. 413–434.
- [5] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Seventh international AAAI conference on weblogs and social media*, 2013.
- [6] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07, 2007.
- [7] E. Elmurungi and A. Gherbi, *Detecting Fake Reviews through Sentiment Analysis Using Machine Learning Techniques*. IARIA/DATA ANALYTICS, 2017.
- [8] V. Singh, R. Piryani, A. Uddin, and P. Waila, "Sentiment analysis of movie reviews and blog posts," in *Advance Computing Conference (IACC)*, 2013, pp. 893–898.
- [9] A. Molla, Y. Biadgie, and K.-A. Sohn, "Detecting Negative Deceptive Opinion from Tweets," in *International Conference on Mobile and Wireless Technology*. Singapore: Springer, 2017.
- [10] S. Shojaee *et al.*, "Detecting deceptive reviews using lexical and syntactic features." 2013.
- [11] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," *Information Sciences*, vol. 385, pp. 213–224, 2017.
- [12] H. Li *et al.*, "Spotting fake reviews via collective positive-unlabeled learning." 2014.
- [13] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ser. WSDM '08, 2008, pp. 219–230.
- [14] 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, no. 2, pp. 456–481, 2016.
- [15] E. D. Wahyuni and A. Djunaidy, "Fake review detection from a product review using modified method of iterative computation framework." 2016.
- [16] D. Michie, D. J. Spiegelhalter, C. Taylor *et al.*, "Machine learning," *Neural and Statistical Classification*, vol. 13, 1994.
- [17] T. O. Ayodele, "Types of machine learning algorithms," in *New advances in machine learning*. InTech, 2010.
- [18] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.
- [19] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features." 1998.
- [20] T. R. Patil and S. S. Sherekar, "Performance analysis of naive bayes and j48 classification algorithm for data classification," pp. 256–261, 2013.
- [21] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.
- [22] N. Suguna and K. Thanushkodi, "An improved k-nearest neighbor classification using genetic algorithm," *International Journal of Computer Science Issues*, vol. 7, no. 2, pp. 18–21, 2010.
- [23] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote sensing of environment*, vol. 61, no. 3, pp. 399–409, 1997.
- [24] A. Liaw, M. Wiener *et al.*, "Classification and regression by random-forest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [25] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
- [26] G. G. Chowdhury, "Natural language processing," *Annual review of information science and technology*, vol. 37, no. 1, pp. 51–89, 2003.
- [27] J. J. Webster and C. Kit, "Tokenization as the initial phase in nlp," in *Proceedings of the 14th conference on Computational linguistics-Volume 4*. Association for Computational Linguistics, 1992, pp. 1106–1110.
- [28] C. Silva and B. Ribeiro, "The importance of stop word removal on recall values in text categorization," in *Neural Networks, 2003. Proceedings of the International Joint Conference on*, vol. 3. IEEE, 2003, pp. 1661–1666.
- [29] J. Plisson, N. Lavrac, D. Mladenic *et al.*, "A rule based approach to word lemmatization," 2004.
- [30] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 4, pp. 388–400, 1993.
- [31] N. Jindal and B. Liu, "Opinion spam and analysis." in *Proceedings of the 2008 international conference on web search and data mining*. ACM, 2008.
- [32] M. Hu and B. Liu, "Mining and summarizing customer reviews." 2004.
- [33] R. Mihalcea, C. Corley, C. Strapparava *et al.*, "Corpus-based and knowledge-based measures of text semantic similarity," in *AAAI*, vol. 6, 2006, pp. 775–780.
- [34] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, 2003, pp. 133–142.
- [35] G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh, "Exploiting burstiness in reviews for review spammer detection," in *Seventh international AAAI conference on weblogs and social media*, 2013.