International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-4, February 2019

Classifiers Ensemble for Fake Review Detection

Harish Baraithiya, R. K. Pateriya

Abstract: ***The growth of e-commerce businesses has attracted many consumers, because they offer a range of products at competitive prices. One thing most purchasers rely on when they purchase online is for product reviews to conclude their decision. Many sellers use the decision to impact the review to hire the paid review authors. These paid review authors target the particular brand, store or product and write reviews to promote or demote them according to the requirements of their hired employees. In view of the effects of these fake reviews, a number of techniques to detect these fake reviews have already been proposed. Because of nature of the reviews it is difficult to classify them using single classifier. Hence in this paper, we proposed an ensemble classifier based approach to detect the fake reviews. The proposed ensemble classifier uses support vector machine (SVM), Naïve Byes classifier and k- nearest neighbor (KNN mutual) classifiers. The proposed technique is evaluated using Yelp and Ott. et al [10] datasets. The evaluation results show that the proposed classifier provides better classification accuracy on both datasets.***

Keywords: ***Fake review detection, ensemble classifiers, Behavioral analysis, Opinion Spam.***

I.INTRODUCTION

The recent growth of online shopping attracted many people, since online shopping sites such as Amazon.com, Flipkart.com, Ebay.com, etc. offer a wide range of attractive products that cannot be easily accessed on the local market. The product information is nevertheless provided on the online store by the seller, which contains very small details in most cases. The customers rely on reviews by other people who have already purchased the product in such cases. The Cone Research Study report [1] reveals the impact of these reviews. This report shows that 4 out of 5 purchasers have changed their choice of purchasing a product after the negative product review, while 9 of 10 purchasers reiterate their decision to purchase after the positive product review. The significance of online product reviews is also investigated in [2], which found that reviews have a tremendous impact on new product sales. The significant impact of reviews on buyer views and product sales offers importers sufficient incentives to manage sales of products through fake reviews. The impostor engages fake review writers, also known as opinion spammers. These spammers write the fake reviews that target a product, brand or service as per the requirements of the impostor. While it is known that it is not easy to detect such review when misleading reviews are present. The first problem is the lack of real datasets for ground truths and the only known close to ground truth datasets, is Yelp.com.

Manuscript published on 28 February 2019.
*****Correspondence Author(s)
Harish Baraithiya, Research Scholar, Department of Computer Science & Engineering, Maulana Azad National Institute of Technology, Bhopal (M.P), India.
Dr. R. K. Pateriya, Associate Professor, Department of Computer Science & Engineering, Maulana Azad National Institute of Technology, Bhopal (M.P), India.

As most detection techniques depend on machine learning, the ground truth data set plays a very important role and its accuracy depends heavily on. In addition, the fact that fake review writing is a dynamic process in which the counterfeit reviewers can improve their writing technique during the course of time to match the non-fake reviews is a major problem. This fake review of the dynamic nature creates difficulties for the classifier trained in a fixed data set. In this work, we present an ensemble classifier based model to deliver efficient review detection system.

II.RELATED WORK

Considering the negative effects of fake reviews a number of techniques have been proposed. Most existing approaches use this as a problem of binary classification [3-10] using different types of classifications, such as the Support Vector Machine (SVM), Artificial Neural Networks (ANN). While the correct selection of classifiers certainly improves performance, studies show [3] that proper selection of features has a much more important influence. A number of features have been proposed and evaluated in view of the impact of features on the accuracy of the classifier. Part of the speech (POS) tags, Linguistic inquiry and word count, which in combination with unigrams and bigrams are used to further improve performance. Jindal et al [17] first addressed the issue which is then followed by a number of other techniques. These techniques can be divided into two main types (texts or metadata review) by learning techniques (supervised, unsupervised and semi-supervised) or feature types. The techniques supervised like Liu et al. [18] used the Bayesian approach and laid out a clustering problem with opinion spam sensing. Mukherjee et al. [4], Chengai et al. [4] and Luyang et al. [9], although all used the Support Vector Machine (SVM) as a classification, are other literature pieces that also have taken supervised learning. Mukherjee et al [18] propose an unsupervised technique based on the Bayesian framework. Other methods are presented in [22] and [23], respectively, such as the relationship- based model (GSRank) and the sectioning of unforeseen rules and rules.

Moving to the features used, content Based techniques use the linguistic characteristics of the review to distinguish false reviews. Part of the language tagging (POS) is one of the most widely used language features. The tagging of POS classifies words used on the basis of word definition and context into noun, pronoun, verb, adverb, adjective etc. The authors of [Ref24] adopted POS tags to sequence patterns to detect gender from the blog writer. For a false review detection also used in [Ref3], POS unigram, POS bigrams, POS sequence patterns, etc.

Many pieces of literatures [3, 5, 10, 14] clearly demonstrate the performance of the POS tags to detect linguistic functions. Another technique for learning emotional, constructive and cognitive components present in writing text samples is a competent and operative approach. Language inquiries and Word Counting (LIWC) The Ott et al. [10] reported that the LIWC has slightly improved on POS in the detection of deceptive opinions spam. The LIWC with simple unigrams and bigrams is also studied in the [3, 9, 15, 25]. Li et al. [25] analyzed a range of disappointing indicators for detecting fake reviews. They also found that the combination of several features such as LIWC or POS allows for greater accuracy in detection. In [7], another language feature called Deep Syntax features appears to be comparable to POS and LIWC as it improves the classification reminder but reduces accuracy. Other word-based features such as word unigrams and bigrams are less effective since they are less effective than POS tags and LIWC in extracting useful linguistic features. Because of the ability of the fake review writer to learn, the fake reviews can be written very similar to real reviews. In these cases, linguistic features can fail only to detect them; therefore a writer may greatly detect fake reviews by additional behavioral information such as the rating pattern, review writing frequencies, etc. Lim et al. [21] presents the pattern of the product ratings and the assessment of spammers. The plunge in the examination pattern to identify a view spammer (some reviews during a short time interval) is presented in [10]. Fake reviews such as those presented in [19, 26] or the text revision [22] are only possible with behavioral information. More information on and among reviewers, who can be used to detect spammer groups presented in [18, 20, 21, 22], is available in the behavioral features. In addition, the behaviors, such as word counts, POS tagging etc, analyze the behavior of reviewer from different angles and do not involve time consuming linguistic analyses.

III.METHODOLOGY

In the section, we discuss the details of ensemble methods, feature extraction techniques frequently used for review text classification, and finally the proposed ensemble classifier model to distinguish deceptive reviews.

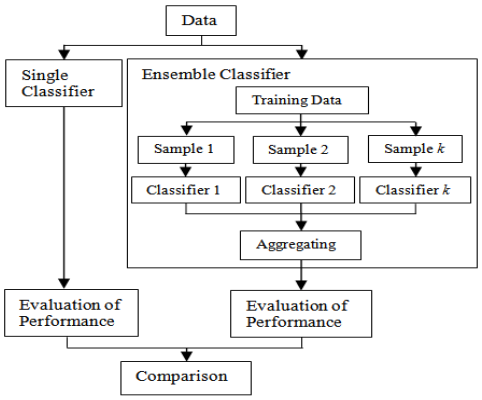

Figure 1: single vs ensemble classifiers, as presented by Utami, et al., 2014 [45].

Ensemble classifier: We consider ensemble methods as a set of techniques of machine learning that combine decisions to improve the performance of the whole system. Similar meanings are also described in the literature as: multiple classifier, multi-strategy apprenticeship, committee, fusion of classifier, combination, aggregated, integrative, etc. In this article, we use ensemble to mention all methods of classification. Voting or weighted voting is the easiest method to combine various learning techniques.

The native idea of ensemble learning means that no single method or arrangement could perform consistently better to any other approach and so the combination of multiple methods will improve the final classifier's performance. Thus, an ensemble classifier performs superior to a single base classifier overall. The success of ensemble approaches depends heavily on the errors made by each different individual basic classifier. The performance of ensemble approaches depends greatly on the precision and diversity characteristics of the basic learners. Several studies have shown that decision-making booms are tending to generate different classifiers in response to minor training variations and thus to make them suited candidates for the basic students of the ensemble system [50, 51, 52]. The easiest approach is to manipulate the training data to generate diverse base classifiers. In this study, we apply the two most common ensemble techniques, bagging and boosting for the purpose of fake review detection.

Input: Training Dataset $< x, y >$, Classification Algorithm CL, Integer N (total iterations)

$For\ i\ =\ 1 \dots N$

$Sample\ a\ subset\ S\ of\ size\ N\ from$
$the\ training\ dataset\ D.$

$End\ for$

Generates a classifier $C(x)$ from S.

Form final classifier $C^*(x)$ by aggregating the j classifiers.

Predict a class label for given data x as follows:

$$C^*(x) = arg \max_y \sum_{i=1}^{N} C_i(x) = y$$

Output: $C^*(x)$.

Figure 2: Pseudo codes for Bagging algorithm.

Bagging: The aim of Bagging was to rearrange the training details, by substituting the original D-training dataset randomly by N elements. Substitution sets are called the replicates for bootstraps where some instances may appear multiple times while others may not. The final $C^*(x)$ classification consists of aggregating $C(x)$ with equal voting for each $C_i(x)$. Figure 2 shows the pseudo codes for bagging algorithm.

Input: Training Dataset $< x, y >$, Classification Algorithm CL, Integer N (total iterations)

Initialize weights w_i for all $x_i \in D$ equal to $1/N$.

$For\ i\ =\ 1 \dots N$

$weighted\ error\ estimate\ err_i$

$$= \frac{\sum w_i\ for\ incorrectly\ classified\ x_i}{\sum_{i=1}^{N} w_i}$$

Compute classifier weight

$$\alpha_i = \frac{1}{2} \log\left(\frac{1 - err_m}{err_m}\right)$$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

For correctly classified instances the weight is

$$w_i = w_i e^{-\alpha_i}$$

For incorrectly classified instances the weight is

$$w_i = w_i e^{\alpha_i}$$

Normalized w_i so that the

$$1 = \sum w_i$$

End for

Form final classifier $C^*(x)$ by a weighted sum of vote of the individual $C_i(x)$ on the basis of its accuracy for detected class.

Output: $C^*(x)$.

Figure 3: Pseudo codes for Boosting algorithm.

Boosting: As another technique to effect training data, at first, each instance was assigned x by the algorithm. The weight is the same. The learning algorithm attempts to minimize the total weighted error in each iteration and returns a $C_i(x)$ classification in each iteration. The weighted $C_i(x)$ error has been calculated and used to update the training instance weights x_i. x_i's weight increases according to its influence on the performance of the classifier, which assigns a greater weight to a miscalculated one and a smaller weight to a correctly classified one. The ultimate classifier $C^*(x)$ is built on the basis of the accuracy of the weighted training set by weighted voting of the individual $C(x)$. The pseudo codes of AdaBoost algorithm is shown in Figure 4.

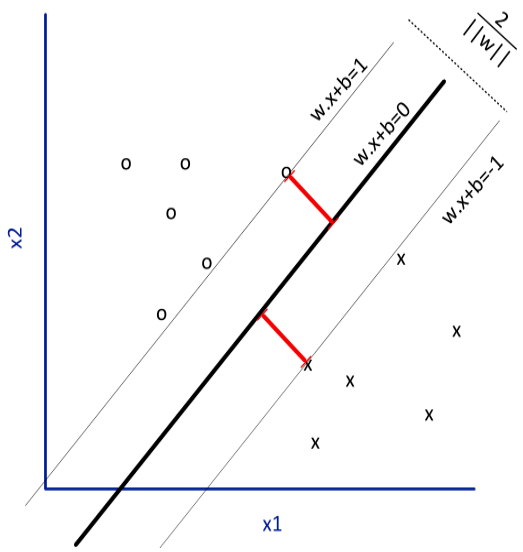

Figure 4: Support Vector Machine [47].

Support Vector Machine (SVM): in its fundamental form it is a binary supervised classification and prediction learning technique [16]. The fundamental objective of the support vector machine is to create a hyperplane or set of hyperplane for separating tuples belonging to two different classes labelled as -1 and + 1. In addition, the longer the hyperplanes distance with the closest data points in training, results in the smaller the error in classification for test data points. Fig. 4 displays two possible hyperplanes separating. A separating hyperplane in mathematical form specified as [48]:

$$\mathbf{W} \cdot \mathbf{x} + \mathrm{b} = \mathbf{0} \quad (1)$$

Where $\mathbf{W} = \{w_1, w_2, \dots, w_n\}$ represents the weight vectors for n features $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$; b is a scalar, and $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ are values of features. For further information on support vector machines, please refer to [48, 49].

K-Nearest Neighbor (KNN): The Nearest Neighbor rule is consistently high performing among the different techniques of statistical pattern recognition, without prejudices about the distributions that the training examples are drawn from. This includes a set of positive as well as negative cases. In calculating the distance to the closest training case, a new sample is classified, and the indication of that point then determines the class of the sample [5, 6]. The k-NN classifier extends this concept by taking the k nearest points and assigning the sign of the majority. In general the value of k is kept smaller and odd to avoid ties. Selecting larger values of k may minimize the effects of noisy data points in the given training data set, and the selection of k is generally done by cross-validation.

Naive Bayes Classifier: The Naive Bayes (Lewis 1992) classification algorithm is called a simple Bayesian algorithm. The categorization of texts has proved to be very effective. With respect to the problem of fake review detection, the document $d \in D$ corresponds to the review instance in which D refers to the set of training views (dataset). Document d can be displayed in terms of features such as word bag, PoS or LIWC. Each feature w to d come from a set of W feature set of all features. A class label (c) is attached to every document (d) where C denotes class label set. The Naive Bayes classifier estimate the $P(c|d)$ probability that a document d belongs to a class c. Through the Bayes rule, the $P(c|d)$ can be given as:

$$P(c|d) \propto P(c) \cdot P(d|c) \quad (2)$$

The crucial consideration of Naive Bayes classifiers is that the features representing the documents are conditionally independent given the class value, hence:

$$P(c|d) \propto P(c) \prod_{w \in d} P(w|c) \quad (3)$$

A general method to estimate $P(w|c)$ is using Laplacian smoothing.

$$P(w|c) = \frac{1 + n(w,c)}{|W| + n(c)} \quad (4)$$

Where $n(w,c)$ is the number of the feature places occupied by w in all training examples which have the class value of c. $n(c)$ is the number of feature places with class value is c. |W| is the entire number of different features in the training set.

IV.EXPERIMENTAL SETUP, RESULTS AND ANALYSIS

We make a comparison of the proposed algorithm and some previously suggested algorithms to evaluate our proposed false revision detection model. This section presents the results of this comparison. The results shown in this section are obtained by averaging the outcomes of five-fold cross validation.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Datasets: We use two various datasets, one known as Amazon Mechanical Turk (AMT), another is known as yelp data sets for the evaluation of the performance of the proposed algorithm. The following are the details of these datasets:

Amazon Mechanical Turk (AMT): is generated by online workers (known as Turkers), by paying one US dollar per review. This dataset includes 400 counterfeit reviews of about 20 Chicago hotels.

Tripadvisor.com: is used to get the 400 non-fake reviews of about 20 hotels as used in AMT.

Some studies show that the AMT dataset is unacceptable since Turkers do not act as a fake professionist. Therefore the experimental results using the AMT dataset are very precise (around 95%).

Yelp.com: While not perfect, it is believed to be almost perfect to detect fake review on the yelp.com website, the fake filtering algorithm. Therefore yelp.com's dataset is supposed to be the basic real dataset. The yelp data set contains about 35k reviews, of which about 14% are classified as fake reviews.

Features: There have been a number of feature extraction techniques for the examination of detections but, in our paper, we evaluated with their bigrams three methods called words counts, POS tags, Linguistic Inquiry and Word Count (LIWC)[10].

Word counts: this is the simplest technique where the text is characterized by a vector of words counts appeared in the text.

POS (part of speech): this technique characterizes the text in terms of part of speech tags such as nouns (N), pronouns (PRP), adverbs (RB), prepositions (IN) etc.

Linguistic Inquiry and Word Count (LIWC): is a common tool for text examination used for social sciences. LIWC software comprises about 4500 words gathered into 80 categories,

N-gram: this technique forms the feature vector from a given text string by counting the arrival of n successive words group.

Limiting Feature vector length: The above features can create a number of features that cause vector sizes to be unnecessarily large. We limit this size by using the features with minimum support value.

Evaluation Criteria: In order to estimate the efficiency of the methods, the classifier are evaluated based on four common measures known as accuracy (eq. 6), precision (eq. 7), recall (eq. 8) and F1 (eq. 5). Accuracy defines the classifier's prediction capacity for fake and non-fake evaluations. Precision defines the predicted label accuracy. Recall defines the classifier's completeness. F1 measurement is the harmonic mean of accuracy and reminder used to balance accuracy with reminder.

The F-measure is defined as:

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5)$$

Accuracy is defined as:

$$Accuracy = \frac{|TP + TN|}{|TP + TN + FP + FN|} \quad (6)$$

Precision is calculated as:

$$Precision = \frac{|TP|}{|TP + FP|} \quad (7)$$

Recall is calculated as:

$$Recall = \frac{|TP|}{|TP + FN|} \quad (8)$$

Where, TP, TN, FP, and FN are representing the true positive, true negative, false positive and false negative.

Two different data sets as presented in Table 3- 4 are used to test and compare the fake review detection performance of the proposed algorithm. The Ott. Dataset is equivalent to 50:50, whereas the data set of yelp, at a ratio of 89:11, is highly uneven.

Table 1: Ott. et el [10] dataset details

Number of fake reviews	400
Number of non-fake reviews	400
Total reviews	800

Table 2: yelp dataset details.

Number of fake reviews	4325
Number of non-fake reviews	34818
Total reviews	39143

To further evaluate the performance of the classifier we experimented with three different features word counts, POS tags, LIWC and their bigrams.

Results Analysis:

Table 3 shows results of previous methods performance assessments as well as the proposed method with Ott. Dataset. When compared to previous methods, the proposed approach provides greater precision for word unigram, word bigram and POS unigram features. For word unigram and bigram features, the proposed classifier works better for almost every measure, except for the recall of non-fake review. The results show that utilization of complex features doesn't improves the performance.

Table 4 for the results: 50:50 yelp dataset distribution. Since the false reviews are written professionally in the yelp data set, it is very difficult to correctly categorize the reviews as the drop in maximum accuracy of approximately 40% is also reflected in the results. However the accuracy by the proposed classifier increased from 55.6% to 57.3%, the precision also increased from 56.8 to 61.8 percent with this dataset and POS-unigram feature. The proposed classifier gives the best accuracy for POS bigram which reaches up to 63.2 percent.

The classification is difficult for the yelp dataset with a natural class distribution (89:11), as the class distribution is highly unequal. Even with this data set, the proposed POS unigram algorithm increases precision to 52.9% from 51.9%; the POS bigram is still the best feature in terms of accuracy.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

V.CONCLUSION

In this paper we presented a classifier ensemble approach for fake review detection by grouping the SVM, KNN and Naïve Byes classifier. The proposed technique ensemble the classifiers to overcome each other's limitations and increases the overall performance.

Finally, two different data sets named Ott and yelp.com are used to estimate the performance of the proposed algorithm. The algorithm is also used to compare its performance with previously proposed algorithms in combination with various feature extraction techniques. The experimental results show that for both datasets and for balanced and unbalanced class distributions the proposed algorithm provides substantial improvement in performance over previous algorithms

Table 3: Performance comparison for AMT dataset.

			Truthful			Deceptive		
Classifier	Feature Type	Accuracy (%)	Precision	Recall	F1	Precision	Recall	F1
SVM Ott. et al [10]	Word Unigram	88.4	89.9	86.5	88.2	87.0	90.3	88.6
NB Ott. et al [10]	Word Unigram	88.4	92.5	83.5	87.8	85.0	93.3	88.9
Proposed Classifier	Word Unigram	**89.1**	86.1	**93.2**	**89.5**	**92.6**	84.9	88.6
SVM Ott. et at [10]	Word Bigram	89.6	90.1	89.0	89.6	89.1	90.3	89.7
NB Ott. et al [10]	Word Bigram	88.9	89.8	87.8	88.7	88.0	90.0	89.0
Proposed Classifier	Word Bigram	**91.0**	90.1	**92.2**	**91.2**	**92.0**	89.9	**90.9**
SVM Ott. et al [10]	POS Unigram	73.0	75.3	68.5	71.7	71.7	77.5	74.2
Proposed Classifier	POS Unigram	**74.0**	70.7	**82.2**	**76.0**	**78.7**	65.9	71.7
Proposed Classifier	POS Bigram	82.5	84.9	79.1	81.9	80.4	85.9	83.1
SVM Ott. et al [10]	LIWC Unigram	76.8	77.2	76.0	76.6	76.4	77.5	76.9
Proposed Classifier	LIWC Unigram	76.0	75.9	**76.1**	76.0	76.0	75.9	75.9
SVM Ott. et al [10]	LIWC Bigram	89.8	89.8	89.8	89.8	89.8	89.8	89.8
Proposed Classifier	LIWC Bigram	89.7	85.8	**95.1**	**90.2**	**94.5**	84.3	89.1

Table 4: Performance comparison for yelp dataset class distribution is kept at ratio 50:50.

			Truthful			Deceptive		
Classifier	Feature Type	Accuracy (%)	Precision	Recall	F1	Precision	Recall	F1
Proposed Classifier	Word Unigram	51.5	51.5	52.0	51.7	51.5	51.0	51.3
Proposed Classifier	Word Bigram	53.3	53.2	55.0	54.1	53.4	51.6	52.5
SVM Mukherjee et al [3]	POS Unigram	55.6	--	--	--	59.5	70.3	64.5
Proposed Classifier	POS Unigram	**57.3**	56.8	61.0	58.8	57.9	53.6	55.7
Proposed Classifier	POS Bigram	63.2	61.8	69.6	65.4	65.2	56.9	60.8
Proposed Classifier	LIWC Unigram	59.7	59.2	62.6	60.8	60.3	56.8	58.5
Proposed Classifier	LIWC Bigram	62.1	61.1	66.4	63.7	63.2	57.8	60.4
SVM Mukherjee et al [3]	W-Bigrams + POS-Bigrams	68.1	--	--	--	65.1	72.4	68.6
SVM Mukherjee et al [3]	W-Bigrams + POS Seq. Pat.	67.7	--	--	--	66.2	74.2	69.9

Table 5: Performance comparison for yelp dataset class distribution is kept at natural ratio (89:11).

			Truthful			Deceptive		
Classifier	Feature Type	Accuracy (%)	Precision	Recall	F1	Precision	Recall	F1
Proposed Classifier	Word Unigram	53.1	53.0	54.4	53.7	53.2	51.8	52.5
Proposed Classifier	Word Bigram	61.1	60.4	64.4	62.3	61.9	57.8	59.8
SVM Mukherjee et al [3]	POS Unigram	51.9	--	--	--	16.9	70.3	27.2
Proposed Classifier	POS Unigram	**52.6**	52.5	54.4	53.4	52.7	50.8	**51.7**
Proposed Classifier	POS Bigram	62.6	62.2	64.4	63.3	63.1	60.8	61.9
Proposed Classifier	LIWC Unigram	53.6	53.5	54.4	53.9	53.6	52.8	53.2
Proposed Classifier	LIWC Bigram	63.6	62.4	68.4	65.3	65.0	58.8	61.8
SVM Mukherjee et al [3]	W-Bigrams + POS-Bigrams	59.6	--	--	--	21.6	72.4	33.3

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

SVM Mukherjee et al [3]	W-Bigrams + POS Seq. Pat.	58.9	--	--	--	20.3	74.2	31.8

REFERENCES

1. Cone Research. Game changer: Corn survey finds 4-out-of-5 consumers reverse purchase decisions based on negative online reviews. 2011. www.conecomm.com/contentmgr/5showdetails.php/id/4008.
2. Cui, Geng and Lui, Hon-Kwong and Guo, Xiaoning, The effect of online consumer reviews on new product sales, International Journal of Electronic Commerce, 17, 39-58 (2012).
3. Mukherjee, Arjun and Venkataraman, Vivek and Liu, Bing and Glance, Natalie, Fake review detection: Classification and analysis of real and pseudo reviews, Technical Report UIC-CS-2013--03, University of Illinois at Chicago, Tech. Rep., (2013).
4. Sun, Chengai and Du, Qiaolin and Tian, Gang, Exploiting product related review features for fake review detection, Mathematical Problems in Engineering, (2016).
5. Dave, Kushal and Lawrence, Steve and Pennock, David M, Mining the peanut gallery: Opinion extraction and semantic classification of product reviews, Proceedings of the 12th international conference on World Wide Web, 519--528 (2003).
6. Duan, Huiying and Zirn, C{\"a}cilia, Can we identify manipulative behavior and the corresponding suspects on review websites using supervised learning?, Nordic Conference on Secure IT Systems, 215--230 (2012).
7. Feng, Song and Banerjee, Ritwik and Choi, Yejin, Syntactic stylometry for deception detection, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 2, 171--175 (2012).
8. Yuan, Ling and Li, Dan and Wei, Shikang and Wang, Mingli, Research of Deceptive Review Detection Based on Target Product Identification and Metapath Feature Weight Calculation, Complexity, (2018).
9. Li, Luyang and Qin, Bing and Ren, Wenjing and Liu, Ting, Document representation and feature combination for deceptive spam review detection, Neurocomputing, 254, 33--41 (2017).
10. Ott, Myle and Choi, Yejin and Cardie, Claire and Hancock, Jeffrey T, Finding deceptive opinion spam by any stretch of the imagination, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 1, 309--319 (2011).
11. Khalifa, Malika Ben and Elouedi, Zied and Lef{\`e}vre, Eric, Fake Reviews Detection Under Belief Function Framework, International Conference on Advanced Intelligent Systems and Informatics, 395--404 (2018).
12. Mukherjee, Subhabrata and Dutta, Sourav and Weikum, Gerhard, Credible review detection with limited information using consistency features, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 195--213 (2016).
13. Chauhan, Shashank Kumar and Goel, Anupam and Goel, Prafull and Chauhan, Avishkar and Gurve, Mahendra K, 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN), , 390--393 (2017).
14. Mukherjee, Arjun and Venkataraman, Vivek and Liu, Bing and Glance, Natalie S, What yelp fake review filter might be doing?, ICWSM, 409--418 (2013).
15. Rout, Jitendra Kumar and Dalmia, Anmol and Choo, Kim-Kwang Raymond and Bakshi, Sambit and Jena, Sanjay Kumar, Revisiting Semi-Supervised Learning for Online Deceptive Review Detection., IEEE Access, 5-1, 1319--1327 (2017).
16. Liu, Yuanchao and Pang, Bo, A Unified Framework for Detecting Author Spamicity by Modeling Review Deviation, Expert Systems with Applications, (2018).
17. Jindal, Nitin and Liu, Bing, Opinion spam and analysis, Proceedings of the 2008 international conference on web search and data mining, 219--230 (2008).
18. Mukherjee, Arjun and Kumar, Abhinav and Liu, Bing and Wang, Junhui and Hsu, Meichun and Castellanos, Malu and Ghosh, Riddhiman, Spotting opinion spammers using behavioral footprints, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 632--640 (2013).
19. Wang, Guan and Xie, Sihong and Liu, Bing and Philip, S Yu, Review graph based online store review spammer detection, 2011 ieee 11th international conference on Data mining (icdm), 1242--1247 (2011).
20. Mukherjee, Arjun and Liu, Bing and Wang, Junhui and Glance, Natalie and Jindal, Nitin, Detecting group review spam, Proceedings of the 20th international conference companion on World wide web. 93--94 (2011).
21. Lim, Ee-Peng and Nguyen, Viet-An and Jindal, Nitin and Liu, Bing and Lauw, Hady Wirawan, Detecting product review spammers using rating behaviors, Proceedings of the 19th ACM international conference on Information and knowledge management, 939--948 (2010).
22. Mukherjee, Arjun and Liu, Bing and Glance, Natalie, Spotting fake reviewer groups in consumer reviews, Proceedings of the 21st international conference on World Wide Web, 191--200 (2012).
23. Jindal, Nitin and Liu, Bing and Lim, Ee-Peng, Finding unusual review patterns using unexpected rules, Proceedings of the 19th ACM international conference on Information and knowledge management, 1549--1552 (2010).
24. Mukherjee, Arjun and Liu, Bing, Improving gender classification of blog authors, Proceedings of the 2010 conference on Empirical Methods in natural Language Processing, 207--217 (2010).
25. Li, Jiwei and Ott, Myle and Cardie, Claire and Hovy, Eduard, Towards a general rule for identifying deceptive opinion spam, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 1, 1566--1576 (2014).
26. Li, Fangtao and Huang, Minlie and Yang, Yi and Zhu, Xiaoyan, Learning to identify review spam, IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 22-3, 2488 (2011).
27. Hammad, A. S. A., and Alaa El-Halees. "An approach for detecting spam in arabic opinion reviews." The International Arab Journal of Information Technology 12 (2013).
28. Fei, Geli, et al. "Exploiting Burstiness in Reviews for Review Spammer Detection." Icwsm 13 (2013): 175-184.
29. Alexander Poliakov "An Example of Fuzzy Decision-Making System", Proceedings of the 7th Korea-Russia International Symposium,KORUS 2003.
30. Taneja, Shweta, Bhawna Suri, Himanshu Narwal, Anchit Jain, Akshay Kathuria, and Sachin Gupta. "A new approach for data classification using Fuzzy logic." In Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference, pp. 22-27. IEEE, 2016.
31. Song, Qinbao, Jingjie Ni, and Guangtao Wang. "A fast clustering-based feature subset selection algorithm for high-dimensional data." IEEE transactions on knowledge and data engineering 25, no. 1 (2013): 1-14.
32. Lin, Lin, Xue-Hai Yuan, and Zun-Quan Xia. "Multicriteria fuzzy decision-making methods based on intuitionistic fuzzy sets." Journal of computer and System Sciences 73, no. 1 (2007): 84-88.
33. Perera, L. P., J. P. Carvalho, and C. Guedes Soares. "Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions." Journal of marine science and technology 16, no. 1 (2011): 84-99.
34. Ishibuchi, Hisao, Ken Nozaki, Naohisa Yamamoto, and Hideo Tanaka. "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms." Fuzzy sets and systems 65, no. 2-3 (1994): 237-253.
35. Zhou, Enwang, and Alireza Khotanzad. "Fuzzy classifier design using genetic algorithms." Pattern Recognition 40, no. 12 (2007): 3401-3414.
36. Ghosh, Soumadip, Sushanta Biswas, Debasree Sarkar, and Partha Pratim Sarkar. "A novel Neuro-fuzzy classification technique for data mining." Egyptian Informatics Journal 15, no. 3 (2014): 129-147.
37. Nauck, Detlef, and Rudolf Kruse. "A neuro-fuzzy method to learn fuzzy classification rules from data." Fuzzy sets and Systems 89, no. 3 (1997): 277-288.
38. Zadeh, Lotfi A. "Fuzzy sets." Information and control 8, no. 3 (1965): 338-353.
39. Mamdani, Ebrahim H., and Sedrak Assilian. "An experiment in linguistic synthesis with a fuzzy logic controller." International journal of man-machine studies 7, no. 1 (1975): 1-13.
40. Yager, Ronald R., and Dimitar P. Filev. "Essentials of fuzzy modeling and control." New York 388 (1994).
41. Ikonomakis, M., Sotiris Kotsiantis, and V. Tampakas. "Text classification using machine learning techniques." WSEAS transactions on computers 4, no. 8 (2005): 966-974.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

42. Ishibuchi, Hisao, and T. Nakashima. "Designing compact fuzzy rule-based systems with default hierarchies for linguistic approximation." In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 3, pp. 2341-2348. IEEE, 1999.
43. Battiti, Roberto. "Using mutual information for selecting features in supervised neural net learning." IEEE Transactions on neural networks 5, no. 4 (1994): 537-550.
44. Quadrianto, Novi, Alex J. Smola, Tiberio S. Caetano, and Quoc V. Le. "Estimating labels from label proportions." Journal of Machine Learning Research 10, no. Oct (2009): 2349-2374.
45. Utami, Iut Tri, Bagus Sartono, and Kusman Sadik. "Comparison of Single and Ensemble Classifiers of Support Vector Machine and Classification Tree." *Journal of Mathematical Sciences and Applications* 2.2 (2014): 17-20.
46. Ektefa, Mohammadreza, Sara Memar, Fatimah Sidi, and Lilly Suriani Affendey. "Intrusion detection using data mining techniques." In *Information Retrieval & Knowledge Management,(CAMP), 2010 International Conference on*, pp. 200-203. IEEE, 2010.
47. Etaiwi, Wael, Arafat Awajan, and Dima Suleiman. "Statistical Arabic Name Entity Recognition Approaches: A Survey." *Procedia Computer Science* 113 (2017): 57-64.
48. J. Han, and M. Kamber, "Data mining: concepts and techniques'" (2nd ed.). Morgan Kaufmann Publishers, 2006.
49. D. L. Olson, D. Delen, "Advanced Data Mining Techniques," 2008.
50. Breiman, Leo. "Bagging predictors." *Machine learning* 24, no. 2 (1996): 123-140.
51. Bauer, Eric, and Ron Kohavi. "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants." *Machine learning* 36, no. 1-2 (1999): 105-139.
52. Dietterich, Thomas G. "Ensemble methods in machine learning." In *International workshop on multiple classifier systems*, pp. 1-15. Springer, Berlin, Heidelberg, 2000.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)