Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-10299-5

Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features

Gregorius Satia Budhi [1,2] · Raymond Chiong [1] · Zuli Wang [3]

Received: 5 December 2019 / Revised: 30 September 2020 / Accepted: 22 December 2020
Published online: 13 January 2021

Abstract
Fraudulent online sellers often collude with reviewers to garner fake reviews for their products. This act undermines the trust of buyers in product reviews, and potentially reduces the effectiveness of online markets. Being able to accurately detect fake reviews is, therefore, critical. In this study, we investigate several preprocessing and textual-based featuring methods along with machine learning classifiers, including single and ensemble models, to build a fake review detection system. Given the nature of product review data, where the number of fake reviews is far less than that of genuine reviews, we look into the results of each class in detail in addition to the overall results. We recognise from our preliminary analysis that, owing to imbalanced data, there is a high imbalance between the accuracies for different classes (e.g., 1.3% for the fake review class and 99.7% for the genuine review class), despite the overall accuracy looking promising (around 89.7%). We propose two dynamic random sampling techniques that are possible for textual-based featuring methods to solve this class imbalance problem. Our results indicate that both sampling techniques can improve the accuracy of the fake review class—for balanced datasets, the accuracies can be improved to a maximum of 84.5% and 75.6% for random under and over-sampling, respectively. However, the accuracies for genuine reviews decrease to 75% and 58.8% for random under and over-sampling, respectively. We also discover that, for smaller datasets, the Adaptive Boosting ensemble model outperforms other single classifiers; whereas, for larger datasets, the performance improvement from ensemble models is insignificant compared to the best results obtained by single classifiers.

Keywords Fake review detection · Textual-based features · Machine learning · Imbalanced data

✉ Raymond Chiong
Raymond.Chiong@newcastle.edu.au

✉ Zuli Wang
wangzuli@cuit.edu.cn

Extended author information available on the last page of the article

1 Introduction

Sellers or vendors these days often engage in fraudulent behaviour to boost the rating for their products, by colluding with reviewers to give good ratings and reviews of their products while providing poor ratings and reviews to their competitors' products [32, 47]. The fraudulent behaviour extends to some companies selling their service to give good reviews and ratings to the sellers or vendors [23, 41]. This malicious practice can also be used to attack competitors by damaging the reputation of their products [28, 64]. The main reason for these fraudulent acts is that consumers trust the opinions expressed in online reviews, and depend on these online reviews for making purchasing decisions [2, 12, 35]. Fraudulent acts resulting in what is commonly called fake or incentivised reviews will render the use of consumer reviews ineffective and undermine the effectiveness of the online market [39]. Unless these reviews are identified, social media will become increasingly filled with lies and deception, and will eventually become completely useless [50].

The occurrence of fake reviews has begun to cause serious concern among some online commerce providers. For example, 30% of Amazon reviews and 52% of reviews posted on Walmart.com were fake and unreliable [46]; approximately one-third of TripAdvisor reviews were fake [4, 19]; about 16% to 20% of Yelp reviews were fake [17, 38]; and hundreds of Facebook groups provided fake review writing services [58]. Some legal actions have already been taken to address these problems [45, 46, 57], and some preventive measures have been installed [4, 38, 46]. However, these actions and preventive measures need a good detector of fake reviews to be effective, because it is almost impossible for anyone to manually read and properly synthesise the large amount of reviews on different online sites [8, 54].

Fake review detection has thus become an increasingly important research topic, especially in the field of natural language processing. With the vital role played by consumer reviews in people's purchasing behaviour [63], the spread of fake reviews can have some detrimental effect and eventually erode consumers' trust in the reviews [36, 39]. The majority of studies on fake review detection in the literature follow either a textual-based featuring approach, a behavioural-based featuring approach, or a combination of both [3, 28]. The behavioural-based featuring approach requires features related to the behaviours of the reviewers, and is therefore dependent on the metadata provided by a particular system. This makes it difficult for results obtained on one system to be applied to another. In this paper, we focus on the textual-based featuring approach, which is more independent of the system, since this approach extracts input features from the review text itself. We use the fake review data from Rayana and Akoglu [49], which contains four fake review datasets taken from the Yelp! Commerce portal (YelpChi Hotel, YelpChi Restaurant, YelpNYC, and YelpZIP). These fake review datasets have been widely used by other researchers (e.g., see [12, 35]), but they are heavily imbalanced; fake reviews comprise only 10% to 13% of total records for each dataset. This phenomenon adds to the difficulty of system training and can lead to false success, which ultimately renders the whole effort useless.

In their work, Rayana and Akoglu implemented a method called SpEagle to distinguish fake reviews from genuine ones [49]. Instead of using a similar approach, in this paper, we apply machine learning classifiers, both single and ensemble models, for fake review detection. Before extracting the features, we implement text preprocessing such as tokenisation, stopword removal, detection of negation words, correction of elongation words, and part of speech (POS) lemmatisation. To extract the features, we use several bag-of-words (BOW) feature extraction methods such as terms frequency (TF), terms frequency-inverse document frequency (TF-IDF), and n-gram words (from unigram to trigram words). We also look into

how to deal with the imbalanced data so that it does not inordinately affect the performance of the system. To overcome this imbalance problem, we propose a dynamic random over and under-sampling approach that can apply the sampling process depending on the current amount of fake and genuine review samples. Based on the ratio setting, over-sampling will increase the amount of minority class samples, while under-sampling will decrease the amount of majority class samples.

This work contributes not only to the relevant research areas but also to online commerce security problems. Theoretically, our work contributes to the literature of both natural language processing and machine learning by providing insights on how to process text materials, extract text features, and detect fraudulent online consumer reviews. Product reviews are an integral part of online commerce, used as guidance by most consumers to make buying decisions [8]. People depend on reviews provided by previous buyers for gauging the quality of products in online e-commerce markets. Reliable and effective detection of fake reviews will increase the trustworthiness of online commerce [10]. At the same time, sellers are using product reviews to evaluate brand perception and consumers' satisfaction levels [21]. For this reason, maintaining the quality of product reviews is essential. Therefore, detection and elimination of fake reviews is a high priority and urgent goal of online commerce portal providers.

The rest of this paper is organised as follows. In Section 2, we review related work on textual-based fake review detection. We then describe, in detail, the design of the fake review detection system and class imbalance problem in Section 3. Experimental results and discussions are presented in Section 4. Finally, we draw conclusion and highlight our future research directions in Section 5.

2 Related work

As mentioned earlier, the majority of studies on fake review detection follow either a textual-based featuring approach, a behavioural-based featuring approach, or a combination of both the approaches [3, 28]. Textual-based featuring focuses on the linguistic features of text, such as words, POS, n-gram, TF, and other linguistic characteristics [20, 27, 50]. In contrast, the behavioural-based featuring approach focuses more on the behaviour of reviewers, such as user identities, reviewed products, total reviews, ratings given and durations of reviews [1, 3, 55].

The behavioural-based approach is entirely dependent on additional information (metadata) provided by the system. Different systems may produce different sets of features, and therefore, any results obtained are applicable only to a particular system and the tested datasets. Some researchers have, therefore, focused their attention on the textual-based featuring approach, because this approach is more independent of the system. Textual-based featuring extracts input features for detection from the review text itself in the form of words or terms. Here, we will discuss research using the textual-based approach dating from 2015 to the present (see Table 1).

A number of researchers have devoted their efforts to searching for proper preprocessing, featuring and detection methods. Several preprocessing and featuring methods, such as BOW, POS tagging, stemming, stopword removal, punctuation mark removal, and n-gram, were investigated by Etaiwi and Naymat [20] for fake review detection using different machine learning algorithms, including the Support Vector Machine (SVM), Naïve Bayes (NB), Decision Trees (DT), Random Forest (RF) and Gradient-Boosted Trees (GBT). Cardoso et al. [12] created some scenarios for experimenting with the textual-based featuring approach

Table 1 An overview of related work

Reference	Year	Preprocessing	Featuring	Algorithms	Domain
Fusilier et al. [27]	2015	Lowercase, punctuation mark and numerical symbol removal	n-gram	PU-learning, NB, SVM	Hotel
Sun et al. [59]	2016	Tokenising	Bigram & trigram	Bagging (2 SVMs, PWCC)	Variety
Etaiwi and Naymat [20]	2017	Stemming, punctuation mark and stopwords removal	BOW, n-gram, POS tagging	NB, SVM, DT, RF, GBT	Hotel
Ren and Ji [50]	2017	–	Continuous BOW	CNN, RNN, GRNN, SVM	Hotel, Restaurant, Doctor
Li et al. [36]	2017	–	n-gram, POS features	SWNN, LSTM, SVM	Hotel, Restaurant, Doctor
Li et al. [37]	2017	Not explained	n-gram, behavioural featuring	PU-learning, SVM, HMM	Restaurant (in China)
Cardoso et al. [12]	2018	Lowercase, non-alphanumeric tokenising	n-gram, TF-IDF	NB, KNN, DT, RF, Rocchio, MDLText	Hotel, Restaurant
Zhang et al. [65]	2018	Stopword removal	POS, skip-gram	DRI-RCNN, SVM, CNN, GRNN	Hotel, Restaurant, Doctor

using the NB, K-Nearest Neighbours (KNN), DT, RF, Rocchio, and Minimum Description Length Text (MDLText). Li et al. [36] implemented n-gram and POS featuring combined with neural network models such as the Convolutional Neural Network (CNN), Sentence Weighted Neural Network (SWNN) and Long Short-Term Memory (LSTM) to detect fake reviews. Ren and Ji explored models such as the CNN, Recurrent Neural Network (RNN), Gated RNN (GRNN) and SVM for fake review detection in review texts [50].

Other researchers have developed new algorithms or modified existing ones to achieve better performance. For example, Sun et al. [59] proposed a new bagging method that contains two types of classifiers, namely the bigram and trigram SVM and Product Word Composition Classifier (PWCC), to improve the detection of fake reviews based on textual features. Zhang et al. [65] modified the CNN to identify deceptive reviews by a Recurrent CNN (DRI-RCNN), which specialises in detecting fake reviews. Fusilier et al. [27] modified the Positive-Unlabelled (PU)-learning technique and used it to improve the performance of deceptive review detection, while taking into consideration the scarcity of deceptive examples. They conducted experiments on three configurations of the following data subsets: positive opinions, negative opinions, and mixed polarity.

It is worth noting that, rather than focusing directly on the text (i.e., words in the text), there are also researchers focusing on properties of the text such as the number of words and sentences, the existence of question and exclamation marks, the number of negative words, and the number of stop words [26, 48, 52, 60]. In this paper, we discuss our efforts on building a fake review detection system based on textual features. We use some proven methods of preprocessing, featuring and machine learning classifiers. We also focus our efforts on overcoming the problem of imbalanced data samples.

3 Methods

We designed a fake review detection system by considering some methods used by other researchers that have achieved good results. For the preprocessing of reviews texts (see Fig. 1), we implemented several methods as follows:

1) **Punctuation and number removal** is a necessary step for a textual-based approach, since the input features of this approach are, in general, words in a BOW list. Therefore, we first needed to remove all components that are not words from the review text.
2) **Word tokenisation** was then applied to break the text into words. This is another necessary step in preprocessing, in which the text is split to be a BOW.

After the two necessary steps above, we carried out three optional steps that can either be activated or not, as follows:

3) **Stopword removal**. Stopword is a term for words such as *the*, *is*, a*t*, *which* and *on*, which are commonly used in English sentences. These words are usually omitted from the text before being used as training examples. Because of their extensive use in any text, these words usually become the most common words and may mislead detectors from recognising keywords or trait words of a group or class.
4) **Word correction** is an essential step in preprocessing. We implemented three methods for word corrections. The idea of word corrections is to reduce the diversity of features.

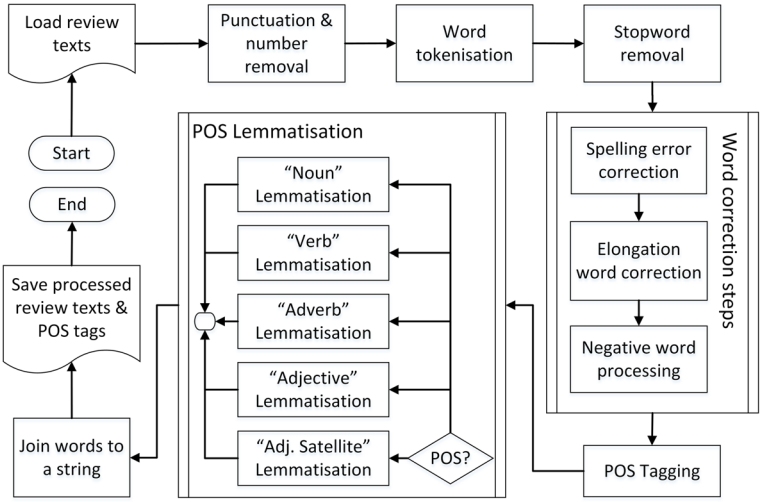

Fig. 1 Preprocessing steps

a. Spelling error correction. Misspelled words add unnecessary difficulty to detection, since the detector recognises them as different words from their correct forms. For this detection and correction of misspelled words, we used Peter Norvig's code for spelling correction [44]. This code is based on probability theory. Here, the investigated word was compared with some words from a large text source, and then the most likely candidate was chosen to replace it.
b. Elongation word correction. Word elongation or word stretchers such as 'yesss', 'fiiine' and 'yoouu' can also increase the diversity of words in the training example and make the classifiers harder to be trained. Therefore, we changed them back to their original form using Peter Norvig's code for spelling correction [44].
c. Detecting negative words. Negative words were replaced with the basic negative word 'not'. This is an important step because negative words change the meaning of a sentence to the opposite of the original meaning. Negative words have many forms, depending on the grammar used in the sentence, but their general purpose is the same: to apply negation to the sentence. Therefore, to reduce diversity, we replaced them with their basic form.

5) **POS tagging and lemmatisation**. POS tagging is a categorical method according to which a word is assigned to its syntactic functions, such as noun, pronoun, adjective, verb and adverb. This step is important because it puts the word in context [20], so that we can lemmatise the word to the correct context during the lemmatisation process. Lemmatisation is a method of changing the word back to its basic form; here, POS lemmatisation returns the chosen word to its basic syntactic functions. This step reduces the diversity of words inside the dataset and makes recognising them easier.

The BOW method is commonly used in textual-based featuring [15]; therefore, we used this method in our work. The BOW method breaks the whole text into a group of singular words. However, since we used it in combination with the n-gram technique, BOW checks the existence of a contiguous sequence of n words from the given sample of text. Here, we limited the n-gram to three (trigrams), since sequences of more than three words were very rare

in real-world texts. After splitting the text into words and trigram terms, we calculated the TF and TF-IDF of each term. Then, we sorted the terms based on their TF or TF-IDF values.

The next step of the system is training machine learning classifiers to be used as the detector. In this step, we implemented and tested three single classifiers and three ensemble algorithms that were successfully used for text recognition and detection [8, 10]. Specifically, we included the following models:

1. Logistic Regression (LR) is a member of the generalised linear model family created by Nelder and Wedderburn in 1972 [42] and improved by Hastie and Tibshirani in 1990 [25]. Traditional linear models are limited to using continuous and normally distributed variables, which is not always desirable. Generalised linear models overcome this problem by using non-normal dependent variables [16, 18]. In LR analysis, the dependent variables can be either unordered polytomous (polytomous nominal) or ordered polytomous (polytomous ordinal), and the independent variables (predictors) can be either interval/ratio variables or dummy variables for representing a limited number of categories [40].
2. SVMs learn from a training dataset and perform classification on unseen data. An SVM separates different classes by a hyperplane and maximises the separation distance as much as possible. The larger the margin, the lower the error generated by the classifier [11]. A linear kernel SVM is generally recommended for text classification [13]; therefore, we used it in our work.
3. The Multilayer Perceptron (MLP) is a supervised learning feedforward artificial neural network. This algorithm computes and updates all the weights in its network to minimise error. It consists of two phases: a feedforward phase, during which the training data is forwarded to the output layer; and the second phase, where the difference between this output and the desired target (we call it the error) is used to update the weights of the network [53]. In this study, we used an improved version of the algorithm [24, 33].
4. The Bagging Predictor (BP) ensemble model uses several single predictors to build a cluster of predictors. These predictors are trained through a bootstrapping process that replicates the training set. Bagging utilises plurality votes to predict a class [5]. In addition to the DT, which is the default base predictor of BP, here we investigated BPs using different single classifiers as the base predictor, namely the LR, SVM and MLP.
5. The RF is an ensemble of DT predictors in which each tree is trained using a random vector independently. Error generalisation of RF depends on the strength of each individual tree and the correlation between them. This ensemble model is relatively robust to outliers and noise [6], and is used in many areas, including text processing [21, 29, 62].
6. The Adaboost (AB) ensemble algorithm iteratively combines multiple weak classifiers over several rounds. It starts with equal weights for all training data. When the training data points are misclassified, the weights of these data points are boosted; then, a new classifier is created using the new unequal weights. This process is repeatedly conducted for a set of classifiers [66].

Our preliminary analysis using the single classifiers showed that overall measurements were good, with results mostly above 80%, while the accuracies and recalls of LR and SVM for YelpNYC were almost 90% (see Table 2). These results are similar to the results obtained by other researchers we discussed in Section 2. However, upon further investigation, it becomes apparent that the accuracy of fake reviews is too low compared with genuine reviews. This is undesirable,

Table 2 Results of preliminary experiments (*)

Dataset	Detector	Overall Result (%)				Detailed Accuracy (%)	
		Acc	Prec	Rec	F1	Fake	Genuine
YelpChi Hotel	LR	83.75	81.15	83.75	82.20	20.92	93.36
	SVM	79.45	80.40	79.45	79.88	28.05	87.34
	MLP	83.45	80.29	83.45	81.55	17.11	93.62
YelpChi Restaurant	LR	85.76	81.45	85.76	82.56	12.94	96.86
	SVM	86.08	81.55	86.08	82.49	11.18	97.50
	MLP	83.90	80.95	83.90	82.14	19.29	93.75
YelpNYC	LR	89.63	84.17	89.63	85.09	1.28	99.72
	SVM	89.74	84.89	89.74	84.95	0.28	99.96
	MLP	85.50	83.41	85.50	84.38	14.65	93.59
YelpZIP	LR	86.80	82.29	86.80	81.54	3.73	99.44
	SVM	86.82	82.45	86.82	81.06	1.60	99.78
	MLP	80.92	79.97	80.92	80.42	22.08	89.87

(*) All of the experiments were conducted using trigrams' BOW combined with TF feature extraction and 10-fold cross-validation.

because we focus on fake review detection not otherwise. The accuracy comparison also showed that measuring only the overall results, without checking each element of the target could lead to false success and ultimately render the effort useless.

In the literature, researchers sometimes claim that the textual-based approach is challenging because there are no significant differences between words in genuine and fake reviews (e.g., see [3]). Researchers also have conflicting opinions about whether the textual-based approach is effective enough in detecting fake reviews in real-world commercial websites [61]. In our opinion, however, the problem may be caused by class imbalance. Table 3 shows that the number of fake reviews is much smaller than that of genuine reviews. This would cause a severe class imbalance issue, which is known to be an impediment to trained machine learning detection of minority samples from majority samples [30, 34]. Machine learning tends to be biased towards majority samples, but it is necessary to classify both majority and minority samples fairly [14, 31].

To check the correctness of our opinion, we added random over-sampling and random under-sampling methods used by Hu et al. [30]. In random over-sampling, we increased the amount of minority class samples (fake reviews) by randomly copying from the existing samples, whereas in random under-sampling we decreased the amount of majority class samples (genuine reviews). We applied the over-sampling process only to the training data

Table 3 Numbers of records in the Yelp fake review datasets [49]

Dataset Name	Total Reviews	Fake Reviews		Genuine Reviews		Total Users	Total Products
		Total	%	Total	%		
YelpChi Hotel	5854	778	13.29	5076	86.71	5026	72
YelpChi Restaurant	61,541	8141	13.23	53,400	86.77	33,037	129
YelpNYC	359,052	36,885	10.27	322,167	89.73	160,225	923
YelpZIP	608,598	80,466	13.22	528,132	86.78	260,277	5044

sections to avoid self-detection of duplicate samples in test data, which can result in overfitting. Moreover, since this work focuses on creating features directly from the review texts, we cannot apply methods that create new synthetic data, such as the Synthetic Minority Oversampling Technique (SMOTE in short) [22] or its variants. The n-fold split and sampling process design is presented in Fig. 2, and the overall system design in Fig. 3.

4 Results and discussion

In this work, we conducted all our experiments using the 10-fold cross-validation method. We used components from Scikit-learn [56] for all the machine learning classifiers included, and also to measure the results. Performance measures considered include accuracy, precision, recall and F-measure (F1), details of which can be found in Table 4. Additionally, we used components from the Natural Language Toolkit [43] and Peter Norvig's code for spelling correction [44] in the preprocessing steps. For all the experiments, we used Yelp's fake review datasets from Rayana and Akoglu [49] (shown in Table 3), and ran them on high performance computing (HPC) facilities provided by the University of Newcastle, Australia. The University's HPC facilities consist of 4000 usable cores for 120 CPU nodes, and 6 GPU nodes, with up to 512 GB RAM and 700 TB of usable shared storage space.

Our experiments focused on the following:

1. Investigating the cost of resources used in preprocessing and the effect of each preprocessing step on fake review detection.
2. Investigating which of TF and TF-IDF is better for extracting features from the fake review texts.
3. Investigating whether random over-sampling and under-sampling can increase the accuracy of the minority class (fake reviews).
4. Investigating which machine learning classifier is the best for detecting fake reviews.

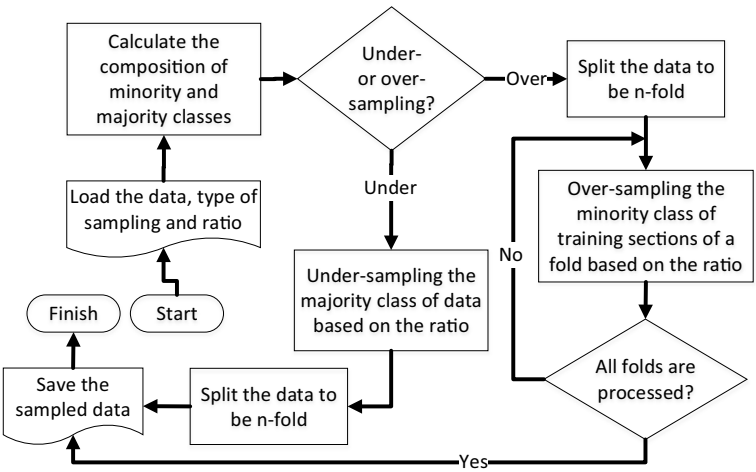

Fig. 2 N-fold split and sampling process design

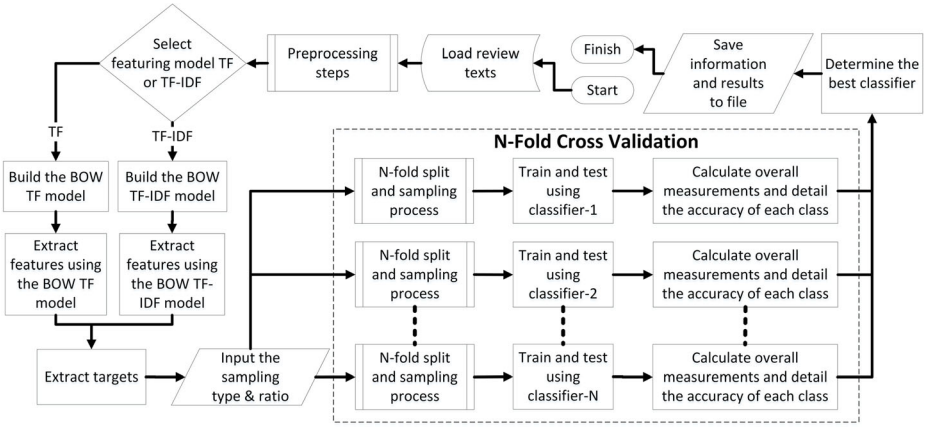

Fig. 3 Overall system design

4.1 Cost calculations of the preprocessing steps

The preprocessing steps sometimes consume more computing resources than expected. Here, we show the cost calculations in terms of processing time, CPU consumption and memory consumption for each preprocessing step. We measured the processing time by capturing the starting and ending times, calculating the difference between them, then dividing the result with total processing samples from Table 3. For calculating the CPU consumption, we used the psutil package [51] to capture the CPU usage percentage of every process, summarised the data, and calculated the average. For memory consumption, we used a standard Python component, namely tracemalloc, to monitor and capture the peak memory usage of the whole process. Results of these cost calculations can be found in Tables 5, 6, and 7, respectively.

As we can see in Table 5, the processing time for each sample is considerably fast. We note from the table that more than 90% of the preprocessing time was spent on word correction.

Table 4 Measurement functions and formulas

No	Name	Sklearn Function	Equation
1	Accuracy	accuracy_score()	$A(y,\widehat{y}) = \frac{1}{n_{samples}} \sum_{k=0}^{n_{samples}-1} 1(\widehat{y_k} = y_k)$ where y is the set of predicted pairs, $\widehat{y}$ is the set of true pairs, and $n_{samples}$ is the total number of samples.
2	Precision	precision_score()	$P = \frac{1}{\sum_{l\in L}\lvert\widehat{y_l}\rvert}\sum_{l\in L}\lvert\widehat{y_l}\rvert P(y_l,\widehat{y_l})$ $P(y_l,\widehat{y_l}) = \frac{tp}{tp+fp}$ where L is the set of classes, y_l is the subset of y with class l, tp is the true positive, and fp is the false positive.
3	Recall	recall_score()	$R = \frac{1}{\sum_{l\in L}\lvert\widehat{y_l}\rvert}\sum_{l\in L}\lvert\widehat{y_l}\rvert R(y_l,\widehat{y_l})$ $R(y_l,\widehat{y_l}) = \frac{tp}{tp+fn}$ where fn is the false negative.
4	F-measure/F1	f1_score()	$F_1 = \frac{1}{\sum_{l\in L}\lvert\widehat{y_l}\rvert}\sum_{l\in L}\lvert\widehat{y_l}\rvert F_1(y_l,\widehat{y_l})$ $F_1(y_l,\widehat{y_l}) = 2*\frac{P(y_l,\widehat{y_l})*R(y_l,\widehat{y_l})}{P(y_l,\widehat{y_l})+R(y_l,\widehat{y_l})}$

Table 5 Average processing time of each preprocessing step

Dataset Name	Average processing time of each sample (ms)			
	Stopword removal	Word correction	POS lemmatisation	All preprocessing
YelpChi Hotel	0.03	0.76	0.03	0.78
YelpChi Restaurant	0.31	9.57	0.32	9.62
YelpNYC	1.52	51.41	1.62	51.42
YelpZIP	2.42	85.71	2.55	85.88

Another observation made is that when the amount of data to be processed increases, the processing time of each sample also increases. We suspect this is because the search time is directly proportional to the number of samples in a dataset.

Calculating CPU consumption of a process is challenging. Many factors could affect the consumption, and it will increase or decrease dynamically based on the process in place. We monitored the CPU usage every time a sample was processed, then calculated the average usage. From the results in Table 6, we can see that our system can optimally utilise the CPU for its preprocessing process. From the results, we note that stopword removal and POS lemmatisation, on average, utilised more CPU time than word correction. However, as we can see in Table 5, word correction needs much more processing time. This is because Peter Norvig's code for spelling correction [44] compares a processed word to a large text source, and then probabilistically decides on the correction from a list of possible candidates.

To measure the memory consumption, we used tracemalloc – a Python component – to detect the peak of memory usage during the process. From Table 7, we can see that the peak memory consumption during each preprocessing step is not very high, considering the amount of data to be processed (see Table 3). This means that a standard personal computer or laptop, which usually would have at least 8GB RAM, should be able to handle the preprocessing steps without any problem.

4.2 Experiments with TF and TF-IDF

In this set of experiments, we investigated whether it is better to use TF or TF-IDF values for creating features using the BOW technique for fake review detection. We used three machine learning models for this comparison: LR, SVM and MLP. When we compared the results of BOW TF-IDF (Table 8) and BOW TF (Table 2), it was clear that using BOW TF-IDF was slightly better overall than using only TF. However, under the Detailed Accuracy column, we

Table 6 Average CPU consumption

Dataset Name	Average CPU consumption (%)			
	Stopword removal	Word correction	POS lemmatisation	All preprocessing
YelpChi Hotel	77.02	78.05	78.02	67.11
YelpChi Restaurant	73.60	59.19	74.35	59.33
YelpNYC	77.88	65.64	78.12	69.43
YelpZIP	75.08	64.99	75.23	72.31

Table 7 Memory consumption

Dataset Name	Peak memory (MB)			
	Stopword removal	Word correction	POS lemmatisation	All preprocessing
YelpChi Hotel	57.82	57.82	124.00	125.74
YelpChi Rest	94.21	103.28	167.41	175.70
YelpNYC	299.59	358.57	390.51	433.19
YelpZIP	447.85	555.19	547.58	638.91

see that most of the accuracy values for fake reviews in TF-IDF are lower than in TF. This means that TF-IDF featuring creates more bias in the detection towards the majority class than TF featuring alone. Since our focus was on increasing the accuracy of the fake review class, we used BOW TF featuring for the subsequent experiments.

4.3 Experiments with random under-sampling and random over-sampling

Over-sampling and under-sampling have their own strengths and weaknesses. The main strength of over-sampling is that it can provide enough data for the minority class, which is essential to train machine learning algorithms. However, random over-sampling creates duplicates that can lead to overfitting. Under-sampling does not create duplicates, since it reduces the size of the majority class. However, the random reduction can also delete some essential traits of the majority class. The other weakness of under-sampling is that, if the minority class is small, it will reduce the number of majority class samples. Theoretically speaking, machine learning algorithms need a large amount of data to perform well.

At first, we tried to balance the data by conducting over-sampling and under-sampling using the ratio 1:1 for fake versus genuine reviews. However, because we found some anomalies in the over-sampling results, we added more experiments for the over-sampling by using additional ratios of 1:2 and 2:1 (see the results in Fig. 4). All experiments in this section were conducted using BOW TF.

Table 8 Results of experiments using TF-IDF

Dataset	Detector	TF-IDF Overall Result (%)				Detailed Accuracy (%)	
		Acc	Prec	Rec	F1	Fake	Genuine
YelpChi Hotel	LR	86.78	81.02	86.78	80.77	0.77	99.96
	SVM	85.80	81.74	85.80	82.78	14.61	96.73
	MLP	84.13	80.98	84.13	82.19	18.62	94.17
YelpChi Restaurant	LR	86.83	82.72	86.83	81.93	5.55	99.22
	SVM	86.57	82.03	86.57	82.28	8.15	98.52
	MLP	84.84	81.41	84.84	82.63	18.08	95.02
YelpNYC	LR	89.72	85.19	89.72	85.09	1.04	99.85
	SVM	89.75	84.76	89.75	84.91	0.05	99.99
	MLP	85.99	83.62	85.99	84.70	5.60	98.88
YelpZIP	LR	86.82	82.44	86.82	81.66	4.21	99.39
	SVM	86.85	83.31	86.85	80.99	1.23	99.87
	MLP	80.78	79.93	80.78	80.33	22.15	89.70

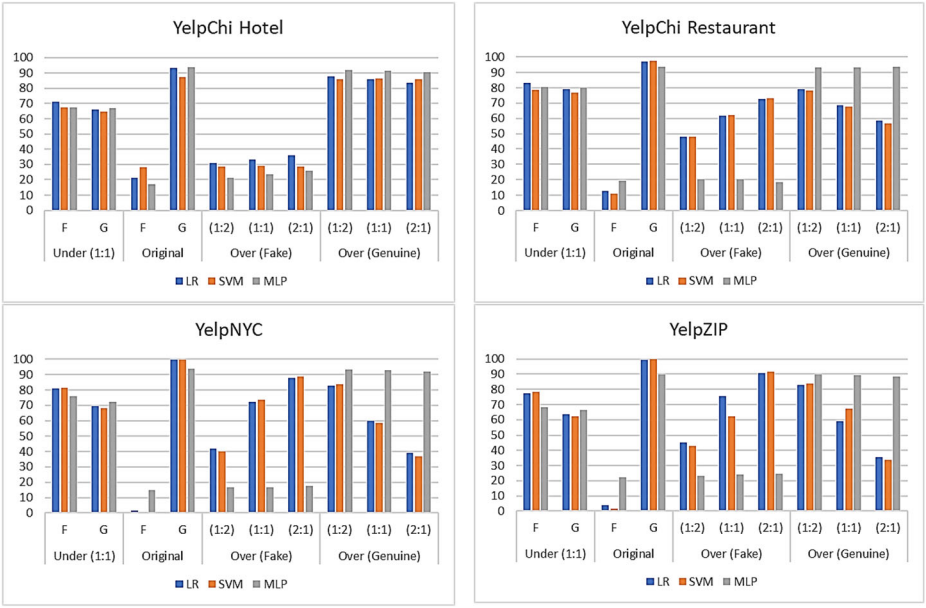

Fig. 4 Accuracy of random under-sampling and over-sampling (in percentage, F = Fake, G = Genuine)

In Fig. 4, it can be seen that both sampling methods for balancing the data can greatly increase the accuracy of the minority class (fake reviews) from almost non-existent in the cases of LR and SVM to more than 60% (see Under (1:1) and Over (1:1)). The increases are even greater in bigger datasets (YelpNYC and YelpZIP).

In the under-sampling experiments, we see that the method can greatly increase the accuracy of the fake review class, but at the same time, it decreases the accuracy of the genuine review class. In our opinion, there are two reasons for this. First, by reducing the number of genuine reviews (majority class), the process of generalisation of both classes in training is balanced. Therefore, the result is also balanced. Second, by reducing the number of genuine samples, some critical traits may not be included in the training step. This makes the genuine review class more difficult to detect.

The over-sampling experiments had some unexpected results. First, the results of YelpChi Hotel did not show as much of an increase in accuracy as did the other datasets. This phenomenon also happened to the MLP-based detector in all datasets (see Fig. 4). To further investigate the phenomenon, we added two other sets of experiments that used 2:1 and 1:2 ratios. The results showed that:

1. In general, when we increased the size of the minority class (using random over-sampling), the detection accuracy of this class increased. This result was expected because we used machine learning classifiers for the detector. From our experience, when utilising machine learning classifiers [7, 8, 30], training them using more samples increases their familiarity with the problem.
2. In the case of YelpChi Hotel, which is the smallest dataset, the increase in accuracy for the fake review class is not as high compared with other datasets. This happens because there are too few samples for the fake review class (see Table 3), so the diversity of this class is low.

3. Although the accuracy still increases when we multiply the existing minority data (fake review class), for the MLP, the increase is not high compared with other algorithms. Owing to the nature of its algorithm training, which backpropagates the error to adjust the weight of neurons inside, presenting duplicate data that is already trained would not produce a significant error to adjust the weight of neurons. This means that providing duplicate data repeatedly in the training phase will not greatly affect accuracy.

Overall, when random over-sampling and under-sampling methods are compared, under-sampling provides better results than over-sampling. We conclude that balancing the classes using the under-sampling method can help increase the accuracy of the minority class more than over-sampling. However, for both sampling methods, we have concerns that increasing the number of minority class samples or reducing the number of majority class samples will decrease the accuracy of the majority class.

4.4 Impact of preprocessing steps on prediction

To further investigate the impact of those optional preprocessing steps on the performance of our prediction system, we conducted additional experiments on all the datasets using the under-sampling setting. We excluded over-sampling because its performance is worse than under-sampling (see Section 4.3). As per the previous section, we ran these experiments using the same machine learning models – LR, SVM and MLP, with BOW-TF featuring and 10-fold cross-validation.

As can be seen in Fig. 5, each of the preprocessing steps shows similar trends on the performance of the prediction system. However, when all of them are combined, the impact becomes clearer, especially on big datasets (YelpNYC and YelpZIP). In small datasets, applying only stopword removal has slightly better results than all preprocessing applied together. The impact of each preprocessing, in general, is higher on small datasets, with the

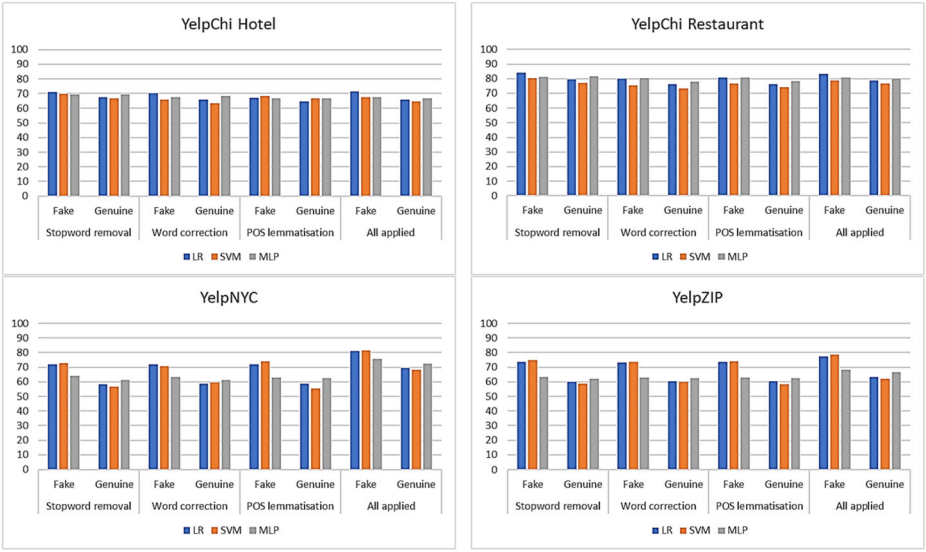

Fig. 5 Accuracy of each preprocessing step for under-sampling prediction (in percentage)

highest impact seen on the YelpChi Restaurant dataset. From this, we can safely assume that restaurant reviews are more homogeneous than other datasets; therefore, the detection is easier. It is worth mentioning here that stopword removal and POS lemmatisation have slightly higher impact on the prediction than word correction, despite word correction taking more than 90% of the preprocessing time (see Table 5).

4.5 Experiments with single and ensemble classifiers

In this section, we compare the performances of all models considered in this work. Our analysis is based only on their performance on the original dataset versus under-sampling. We excluded over-sampling because its performance is worse than under-sampling (see Section 4.3).

As is evident in Fig. 4 (under-sampling) and Fig. 5, LR gives the best results on small datasets (YelpChi Hotel and Restaurant), whereas the SVM produces better results on the bigger datasets (YelpNYC and YelpZIP). While the performance of MLP is worse than that of the other two, it provides more balanced accuracy results for fake and genuine classes.

Besides single machine learning models, we also investigated the performance of ensemble algorithms, namely the RF, BP and AB. For the BP model, we performed experiments by replacing its default base predictor, the DT, with either the LR, SVM or MLP. This idea was inspired by our previous research, in which the BP performed better if its base predictor was replaced with one of these single classifiers [8, 9].

As seen in Fig. 6, under-sampling works well with the ensemble models. It can increase the accuracy of the minority class without significantly decreasing the accuracy of the majority class. AB performs the best when it is used to detect fake reviews in small datasets (YelpChi

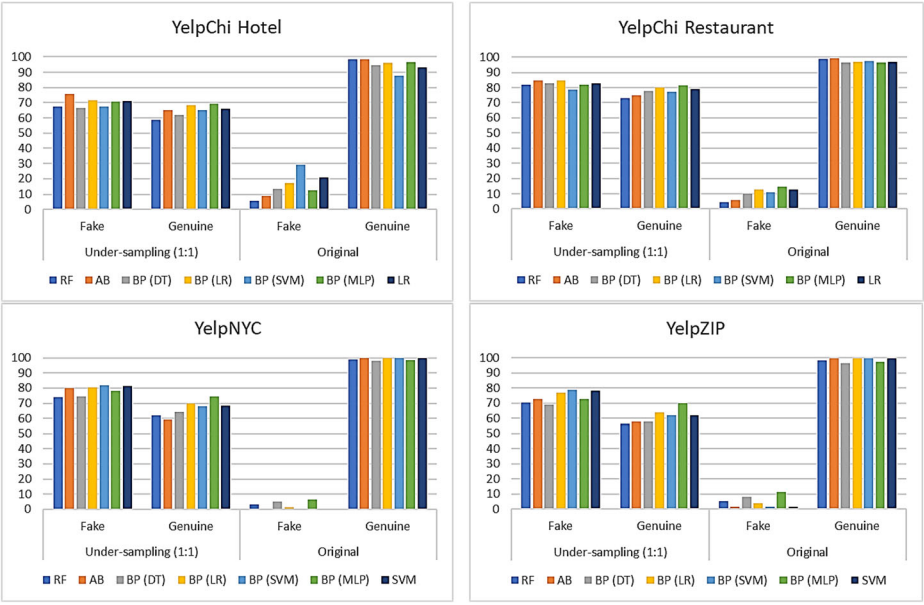

Fig. 6 Accuracy of ensemble models compared to the best single detector of each dataset (in percentage)

Hotel and Restaurant). The increase is quite high when we compare it to the best single classifier for these datasets, which is LR. The increase is 4.5% in YelpChi Hotel and 1.5% in YelpChi Restaurant datasets. For the bigger datasets (YelpNYC and YelpZIP), the BP(SVM) combination performs the best. However, the increase in accuracy compared to the best single classifier, which is the SVM itself, is small (less than 0.5%). Therefore, for the bigger datasets, such as YelpNYC and YelpZIP, a single SVM can be the best choice because it requires less time for training. Another interesting finding is that the BP(MLP) combination can achieve more balanced results in all datasets compared with other ensembles. This BP(MLP) combination can also increase the performance of a single MLP while still maintaining the balance of accuracies between classes.

5 Conclusion

The problem of fake or fraudulent reviews in online commerce today is acute and has prompted companies and researchers to make concerted efforts to find solutions. However, the relative scarcity of fake review sample data makes research in this area challenging. In this paper, we demonstrated that achieving good overall results (i.e. 89.7% accuracy) does not mean the effort is a success. Further investigation revealed that the accuracies of fake and genuine classes are heavily imbalanced; fake review class detection accuracies are between 1% and 28%, as compared with 87% to 99% for the genuine review class. We suspect the imbalanced data samples cause these imbalanced results.

Using random sampling methods, we can overcome the imbalance problem mentioned above. While over-sampling the minority class (fake reviews) can increase the accuracy of the class greatly, this method does not perform well for the smallest dataset (YelpChi Hotel). An interesting finding on preprocessing is that word correction, which consumes more than 90% of the processing time, has less impact on the prediction compared to the other preprocessing steps. However, when all of the preprocessing steps were applied to the datasets, they can improve the performance of fake review prediction more than if each was applied alone, especially in the case of bigger datasets.

We also found that the MLP, which performs best for the original dataset, does not perform well when applied to over-sampled datasets. Otherwise, under-sampling the majority or genuine class can increase the accuracy of the fake review class for any datasets using any classifiers. LR performs best on small datasets, whereas the linear-kernel SVM is the best for bigger datasets. Although the MLP does not produce the best results under the under-sampling method, it provides more balanced accuracies between classes compared with other classifiers. The AB ensemble classifier can further increase the accuracy of the fake review class for small datasets. For bigger datasets, however, the ensembles do not lead to a substantial increase in accuracy compared with the results of the best single classifier. From the results of under-sampling experiments, we can conclude that, given similar numbers of data, fake reviews are more easily detected than genuine reviews. For future work, we plan to investigate the effect of imbalanced datasets on behavioural featuring and the possibility of improving it using a novel ensemble model.

Acknowledgements The first author would like to acknowledge financial support from the Indonesian Endowment Fund for Education (LPDP), Ministry of Finance, and the Directorate General of Higher Education (DIKTI), Ministry of Education and Culture, Republic of Indonesia.

References

1. Akram AU, Khan HU, Iqbal S, Iqbal T, Munir EU, Shafi M (2018) Finding rotten eggs: a review spam detection model using diverse feature sets. KSII Trans Internet Inform Syst 12(10):5120–5142. https://doi.org/10.3837/tiis.2018.10.026
2. Bajaj S, Garg N, Singh SK (2017) A novel user-based spam review detection. Procedia Comput Sci 122: 1009–1015
3. Barbado R, Araque O, Iglesias CA (2019) A framework for fake review detection in online consumer electronics retailers. Inf Process Manag 56(4):1234–1244. https://doi.org/10.1016/j.ipm.2019.03.002
4. Birchall G (2018) TripAdvisor denies claims one in three reviews 'faked'. https://www.news.com.au/technology/online/social/tripadvisor-denies-claims-one-in-three-reviews-faked/news-story/55243de188cc7f1fb2abb52fee3bac45. Accessed October 03 2019
5. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140. https://doi.org/10.1007/bf00058655
6. Breiman L (2001) Random forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/a:1010933404324
7. Budhi GS, Adipranata R (2014) Java characters recognition using evolutionary neural network and combination of Chi2 and backpropagation neural network. Int J Appl Eng Res 9(22):18025–18036
8. Budhi GS, Chiong R, Pranata I, Hu Z (2017) Predicting rating polarity through automatic classification of review texts. In: Proceedings of the 2017 IEEE Conference on Big Data and Analytics, Kuching, Malaysia, pp 19–24. https://doi.org/10.1109/ICBDAA.2017.8284101
9. Budhi GS, Chiong R, Hu Z, Pranata I, Dhakal S (2018) Multi-PSO based classifier selection and parameter optimisation for sentiment polarity prediction. Proceedings of the 2018 IEEE Conference on Big Data and Analytics, Langkawi Island, Malaysia, pp 68–73. https://doi.org/10.1109/ICBDAA.2018.8629593
10. Budhi GS, Chiong R, Pranata I, Hu Z (2020) Using machine learning to predict the sentiment of online reviews: a new framework for comparative analysis. Arch Computation Methods Eng. https://doi.org/10.1007/s11831-020-09464-8
11. Campbell C, Ying Y (2011) Learning with support vector machines. Morgan & Claypool
12. Cardoso EF, Silva RM, Almeida TA (2018) Towards automatic filtering of fake reviews. Neurocomputing 309:106–116. https://doi.org/10.1016/j.neucom.2018.04.074
13. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27. https://doi.org/10.1145/1961189.1961199
14. Darzi MRK, Niaki STA, Khedmati M (2019) Binary classification of imbalanced datasets: the case of CoIL challenge 2000. Expert Syst Appl 128:169–186. https://doi.org/10.1016/j.eswa.2019.03.024
15. Deng X, Li Y, Weng J, Zhang J (2019) Feature selection for text classification: a review. Multimed Tools Appl 78(3):3797–3816. https://doi.org/10.1007/s11042-018-6083-5
16. Dobson AJ, Barnett AG (2008) An introduction to generalized linear models, 3rd edn. CRC Press, Boca Raton
17. D'Onfro J (2013) A whopping 20% of Yelp reviews are fake. https://www.businessinsider.com.au/20-percent-of-yelp-reviews-fake-2013-9). Accessed Oktober 02 2019
18. Dunteman GH, Ho M-HR (2011) Generalized Linear Models. In: An introduction to generalized linear models. SAGE Publications, Inc., pp 2–6
19. Ellson A (2018) A third of TripAdvisor reviews are fake as cheats buy five stars. The Times. https://www.thetimes.co.uk/article/hotel-and-caf-cheats-are-caught-trying-to-buy-tripadvisor-stars-027fbcwc8. Accessed Oktober 02 2019
20. Etaiwi W, Naymat G (2017) The impact of applying different preprocessing steps on review spam detection. Procedia Comput Sci 113:273–279. https://doi.org/10.1016/j.procs.2017.08.368
21. Felbermayr A, Nanopoulos A (2016) The role of emotions for the perceived usefulness in online customer reviews. J Interact Mark 36:60–76. https://doi.org/10.1016/j.intmar.2016.05.004
22. Fernandez A, Garcıa S, Chawla FHNV (2018) SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 61:863–905
23. Freeman LL (2016) How to spot fake online reviews. Money 45(6):30–30
24. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, pp 249–256
25. Hastie T, Tibshirani R (1990) Generalized additive models. Chapman and Hall/CRC,
26. Hazim M, Anuar NB, Ab Razak MF, Abdullah NA (2018) Detecting opinion spams through supervised boosting approach. PLoS One 13(6):e0198884. https://doi.org/10.1371/journal.pone.0198884
27. Hernández Fusilier D, Montes-y-Gómez M, Rosso P, Guzmán Cabrera R (2015) Detecting positive and negative deceptive opinions using PU-learning. Inf Process Manag 51(4):433–443. https://doi.org/10.1016/j.ipm.2014.11.001

28. Heydari A, Ma T, Salim N, Heydari Z (2015) Detection of review spam: a survey. Expert Syst Appl 42(7): 3634–3642. https://doi.org/10.1016/j.eswa.2014.12.029
29. Hu Z, Chiong R, Pranata I, Susilo W, Bao Y (2016) Identifying malicious web domains using machine learning techniques with online credibility and performance data. In: Proceedings of the IEEE Congress on Evolutionary Computation, Vancouver, Canada, pp 5186–5194. https://doi.org/10.1109/CEC.2016.7748347
30. Hu Z, Chiong R, Pranata I, Bao Y, Lin Y (2019) Malicious web domain identification using online credibility and performance data by considering the class imbalance issue. Ind Manag Data Syst 119(3): 676–696. https://doi.org/10.1108/IMDS-02-2018-0072
31. Imran M, Latif S, Mehmood D, Shah MS (2019) Student academic performance prediction using supervised learning techniques. Int J Emerg Technol Learn 14(14):92–104. https://doi.org/10.3991/ijet.v14i14.10310
32. Ivanova O, Scholz M (2017) How can online marketplaces reduce rating manipulation? A new approach on dynamic aggregation of online ratings. Decis Support Syst 104:64–78. https://doi.org/10.1016/j.dss.2017.10.003
33. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations. San Diego, USA, pp 1–15
34. Ko T, Lee JH, Cho H, Cho S, Lee W, Lee M (2017) Machine learning-based anomaly detection via integration of manufacturing, inspection and after-sales service data. Ind Manag Data Syst 117(5):927–945. https://doi.org/10.1108/imds-06-2016-0195
35. Kumar N, Venugopal D, Qiu L, Kumar S (2018) Detecting review manipulation on online platforms with hierarchical supervised learning. J Manag Inf Syst 35(1):350–380. https://doi.org/10.1080/07421222.2018.1440758
36. Li L, Qin B, Ren W, Liu T (2017) Document representation and feature combination for deceptive spam review detection. Neurocomputing 254:33–41. https://doi.org/10.1016/j.neucom.2016.10.080
37. Li H, Fei G, Wang S, Liu B, Shao W, Mukherjee A, Shao J (2017) Bimodal distribution and co-bursting in review spam detection. In: Proceedings of the 26th International Conference on World Wide Web. Perth, Australia, pp 1063–1072. https://doi.org/10.1145/3038912.3052582
38. Luca M, Zervas G (2016) Fake it till you make it: reputation, competition, and yelp review fraud. Manag Sci 62(12):3412–3427. https://doi.org/10.1287/mnsc.2015.2304
39. Malbon J (2013) Taking fake online consumer reviews seriously. J Consum Policy 36(2):139–157. https://doi.org/10.1007/s10603-012-9216-7
40. Menard S (2010) Logistic regression: from introductory to advanced concepts and applications. SAGE, Los Angeles
41. Munzel A (2016) Assisting consumers in detecting fake reviews: the role of identity information disclosure and consensus. J Retail Consum Serv 32:96–108. https://doi.org/10.1016/j.jretconser.2016.06.002
42. Nelder JA, Wedderburn RWM (1972) Generalized linear models. J R Stat Soc Ser A 135(3):370–384. https://doi.org/10.2307/2344614
43. NLTK (2019) Nltk Package. http://www.nltk.org/api/nltk.html. Accessed 25 Jan 2019
44. Norvig P (2016) How to write a spelling corrector. https://norvig.com/spell-correct.html. Accessed June 01 2018
45. O'Neill S (2018) A peddler of fake reviews on TripAdvisor gets jail time. https://skift.com/2018/09/12/fake-reviews-tripadvisor-jail-italy/. Accessed October 03 2019
46. Picchi A (2019) Buyer beware: scourge of fake reviews hitting Amazon, Walmart and other major retailers. CBS News. https://www.cbsnews.com/news/buyer-beware-a-scourge-of-fake-online-reviews-is-hitting-amazon-walmart-and-other-major-retailers/. Accessed 2 Oct 2019
47. Rahman M, Carbunar B, Ballesteros J, Chau DH (2015) To catch a fake: curbing deceptive yelp ratings and venues. Statistic Anal Data Min 8(3):147–161. https://doi.org/10.1002/sam.11264
48. Rathore S, Loia V, Park JH (2018) SpamSpotter: an efficient spammer detection framework based on intelligent decision support system on Facebook. Appl Soft Comput 67:920–932. https://doi.org/10.1016/j.asoc.2017.09.032
49. Rayana S, Akoglu L (2015) Collective opinion spam detection: Bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, pp 985–994. https://doi.org/10.1145/2783258.2783370
50. Ren Y, Ji D (2017) Neural networks for deceptive opinion spam detection: an empirical study. Inf Sci 385-386:213–224. https://doi.org/10.1016/j.ins.2017.01.015
51. Rodola G (2020) psutil 5.7.2. https://pypi.org/project/psutil/. Accessed August 5 2020
52. Rout JK, Singh S, Jena SK, Bakshi S (2016) Deceptive review detection using labeled and unlabeled data. Multimed Tools Appl 76(3):3187–3211. https://doi.org/10.1007/s11042-016-3819-y

53. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol 1. MIT Press, pp 318–362
54. Salehan M, Kim DJ (2016) Predicting the performance of online consumer reviews: a sentiment mining approach to big data analytics. Decis Support Syst 81:30–40. https://doi.org/10.1016/j.dss.2015.10.006
55. Savage D, Zhang X, Yu X, Chou P, Wang Q (2015) Detection of opinion spam based on anomalous rating deviation. Expert Syst Appl 42(22):8650–8657. https://doi.org/10.1016/j.eswa.2015.07.019
56. Scikit-learn (2019) API Reference. https://scikit-learn.org/stable/modules/classes.html. Accessed 19 Mar 2019
57. Shu C (2019) FTC brings its first case against fake paid reviews on Amazon. https://techcrunch.com/2019/02/26/ftc-brings-its-first-case-against-fake-paid-reviews-on-amazon/. Accessed October 03 2019
58. Smithers R (2019) Facebook still flooded with fake reviews, says which? The Guardian. https://www.theguardian.com/business/2019/aug/06/facebook-fake-reviews-which. Accessed October 03 2019
59. Sun C, Du Q, Tian G (2016) Exploiting product related review features for fake review detection. Math Probl Eng 2016:1–7. https://doi.org/10.1155/2016/4935792
60. Wahyuni ED, Djunaidy A (2016) Fake review detection from a product review using modified method of iterative computation framework. Proceed MATEC Web Confer 58:03003. https://doi.org/10.1051/matec
61. Wang X, Liu K, Zhao J (2017) Handling cold-start problem in review spam detection by jointly embedding texts and behaviors. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, pp 366–376. https://doi.org/10.18653/v1/P17-1034
62. Wu Q, Ye Y, Zhang H, Ng MK, Ho SS (2014) ForesTexter: an efficient random forest algorithm for imbalanced text categorization. Knowl-Based Syst 67:105–116. https://doi.org/10.1016/j.knosys.2014.06.004
63. Wu Y, Ngai EWT, Wu P, Wu C (2020) Fake online reviews: literature review, synthesis, and directions for future research. Decis Support Syst 132:113280. https://doi.org/10.1016/j.dss.2020.113280
64. Zhang D, Zhou L, Kehoe JL, Kilic IY (2016) What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fake online reviews. J Manag Inf Syst 33(2):456–481. https://doi.org/10.1080/07421222.2016.1205907
65. Zhang W, Du Y, Yoshida T, Wang Q (2018) DRI-RCNN: an approach to deceptive review identification using recurrent convolutional neural network. Inf Process Manag 54(4):576–592. https://doi.org/10.1016/j.ipm.2018.03.007
66. Zhu J, Zou H, Rosset S, Hastie T (2009) Multi-class AdaBoost. Stat Interface 2(3):349–360. https://doi.org/10.4310/SII.2009.v2.n3.a8

Affiliations

Gregorius Satia Budhi [1,2] · Raymond Chiong [1] · Zuli Wang [3]

[1] School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia

[2] Informatics Department, Petra Christian University, Surabaya 60236, Indonesia

[3] School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China