Electronic Commerce Research and Applications 47 (2021) 101048

Contents lists available at ScienceDirect

Electronic Commerce Research and Applications

journal homepage: www.elsevier.com/locate/elerap

Using a hybrid content-based and behaviour-based featuring approach in a parallel environment to detect fake reviews

Gregorius Satia Budhi [a,b], Raymond Chiong [a,c,*], Zuli Wang [d], Sandeep Dhakal [a]

[a] School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia
[b] Informatics Department, Petra Christian University, Surabaya 60236, Indonesia
[c] School of Economics and Management, Fuzhou University, Fuzhou 350116, China
[d] School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China

ARTICLE INFO

Keywords:
Fake review detection
Featuring approach
Machine learning
Deep learning
Imbalanced data
Parallel processing

ABSTRACT

The financial impact of positive reviews has prompted some fraudulent sellers to generate fake product reviews for either promoting their products or discrediting competing products. Many e-commerce portals have implemented measures to detect such fake reviews, and these measures require excellent detectors to be effective. In this work, we propose 133 unique features from the combination of content and behaviour-based features to detect fake reviews using machine learning classifiers. Preliminary results show that these features can provide good results for all datasets tested. Detailed analysis of the results, however, reveals the existence of class imbalance issues for two of the bigger datasets - there is a high imbalance between the accuracies of different classes (e.g., 7.73% for the fake class and 99.3% for the genuine class using a Multilayer Perceptron classifier). We therefore introduce two sampling methods that can improve the accuracy of the fake review class on balanced datasets. The accuracies can be improved to a maximum of 89% for both random under and over-sampling on Convolutional Neural Networks. Additionally, we propose a parallel cross-validation method that can speed up the validation process in a parallel environment.

1. Introduction

It is common practice for e-commerce portals to allow their customers to write product reviews for their purchases (Utz et al., 2012; Bagheri et al., 2013). Customers' reviews not only will influence their own social circle, but also allow new customers to form their opinion of the product (Bajaj et al., 2017; Budhi et al., 2017). Products with positive reviews from previous purchasers can easily attract potential customers, whereas negative reviews will detract potential buyers. For example, when someone sees that most of the reviews for a product are positive, their intention to buy it will increase. They will, however, look for alternatives when most of the reviews are negative (Feng and Hirst, 2013; Jindal and Liu, 2008). Product reviews are an integral part of online commerce, used as guidance by most customers to make buying decisions (Budhi et al., 2017; Song et al., 2020) and by sellers to evaluate brand perception and customer satisfaction levels (Felbermayr and Nanopoulos, 2016; Budhi et al., 2017). Therefore, maintaining the quality of product reviews is important.

Given the financial benefit associated with reviews, some fraudulent sellers and service providers attempt to manipulate their customers by using fake positive reviews to promote their products and services and inflate potential buyers' belief that previous buyers are pleased with their purchases; fake negative reviews can also be used to dissuade potential customers from competing products and services (Feng and Hirst, 2013; Mukherjee et al., 2013). A fake review, in this context, is a review with fictional opinions written for a commercial motive but promoted as authentic (Li et al., 2017). There is great potential for fake reviews to distort the real evaluation of a product (Feng and Hirst, 2013), erode trust in consumer reviews and eventually undermine the effectiveness of online markets (Malbon, 2013). Unless such reviews are detected and acted upon, social media will be increasingly flooded with lies and deception, and eventually become useless from an e-commerce perspective (Ren and Ji, 2017).

With the intention of curbing this problem, some social media networks allow users to report potential fake or spam reviews (Cardoso et al., 2018). Some e-commerce portals, such as Amazon, Walmart, and TripAdvisor, have already taken legal actions to address this problem (Picchi, 2019; Shu, 2019; O'Neill, 2018). Yelp.com even launched an

* Corresponding author at: School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia.
E-mail address: Raymond.Chiong@newcastle.edu.au (R. Chiong).

https://doi.org/10.1016/j.elerap.2021.101048
Received 14 March 2020; Received in revised form 28 January 2021; Accepted 26 March 2021
Available online 31 March 2021

operation to publicly shame people or companies who used fake reviews (Mukherjee et al., 2013). Several e-commerce portals have also installed preventive measures against these deceptive actions (Picchi, 2019; Luca and Zervas, 2016; Birchall, 2018). Such preventive measures need a good detector algorithm to be effective, since humans find it difficult to detect fake reviews that have deliberately been written like genuine reviews (Li et al., 2017; Cardoso et al., 2018; Ott et al., 2011). In addition, manually reading or properly synthesising the huge number of reviews on e-commerce platforms is almost impossible (Budhi et al., 2017; Salehan and Kim, 2016).

Fake review detection has thus become an urgent and meaningful task in natural language processing studies. With the continuous growth and importance of consumer reviews (Fang et al., 2019), any proliferation of fake reviews would attract attention and likely erode trust in consumer reviews (Li et al., 2017). Most of the studies in fake review detection, to date, utilise two different approaches: 1) based on the content of the reviews, or 2) based on the behaviour of the reviewers. Some studies have also employed a combination of both approaches (e.g., see (Barbado et al., 2019; Heydari et al., 2015). Content-based approaches extract features from linguistic characteristics of the text, such as words, part of speech (POS), and n-gram, term frequency (TF) (Ren and Ji, 2017; Hernández Fusilier et al., 2015; Etaiwi and Naymat, 2017), while behaviour-based approaches extract features from the identity and behaviour of the reviewers, such as total reviews, the number of reviewed products, ratings given, and time of reviews (Savage et al., 2015; Barbado et al., 2019; Akram et al., 2018).

In this paper, we focus our research on the combination of textual content and user behaviour features. Regarding content-based features, we build the features from the characteristics of the review content, such as the number of words, number of sentences, number of questions, number of exclamations, POS, linguistic traits, spam terms, and sentiment terms. Similarly, we extract reviewer-based features from the behaviour of the reviewers, such as the total reviews, duration of reviewer tenure, average duration between reviews, ratings, and so on. In addition to implementing features based on textual content and user behaviour, as is the common practice in the literature, we also incorporate behaviour-based features from the products' perspective. Hence, we are able to capture behaviours from both the reviewers and products.

We use the Yelp fake review datasets from Rayana and Akoglu (Rayana and Akoglu, 2015) in our experiments. There are four datasets (YelpChi Hotel, YelpChi Restaurant, YelpNYC and YelpZIP), ranging from a small to large number of fake reviews, and they have been widely used in the literature (You et al., 2018; Yuan et al., 2019, 2018; Rastogi et al., 2020; Tang et al., 2020). While Rayana and Akoglu implemented SpEagle to detect fake reviews (Rayana and Akoglu, 2015), we apply machine/deep learning classifiers (both single and ensemble models) for fake review detection. To train and test these classifiers, we build specialised feature extraction methods that extract features mainly from linguistic characteristics of the text review in addition to the behaviour of reviewers. We also show how to deal with imbalanced data so that it does not inordinately affect the performance of the classifiers. To speed up the process of investigation, we also design and implement a parallel version of n-fold cross-validation (CV).

The rest of this paper is organised as follows. In the next section, we review related work on fake review detection. After that, we explain the datasets used and the design of features, and provide a detailed description of the fake review detection procedure, class imbalance problem, and parallel CV. Experimental results and discussions are then presented. Finally, we conclude and highlight our future research directions.

2. Related work

In the previous section, we briefly introduced the two main approaches to conducting research on fake review detection. Creating features from the content of the review is generally independent of the system, whereas creating features from the reviewers' behaviours is dependent on the type of data provided by the system. In this section, we discuss latest research using both approaches, dating from 2015 to the present (see Table 1).

The content-based approach can be further categorised into two sub-approaches. The first is to create features from the review text itself using methods such as Bag of Words (BOW), Word2Vec, skip-gram and so on. Thus, the features for detection are words or terms from the review text itself. The features extracted by this approach (hereafter called textual-based featuring) are similar, mostly in the form of n-gram terms (Li et al., 2017; Cardoso et al., 2018; Hernández Fusilier et al., 2015; Etaiwi and Naymat, 2017; Sun et al., 2016). Some produce different types of features, such as Continuous BOW (CBOW) (Ren and Ji, 2017) and skip-gram (Zhang et al., 2018). This textual-based approach is a promising feature extraction approach for text mining in general, and is used by a majority of researchers for directly extracting features from the text. As we can see in Table 1, most studies that used text-only datasets (without additional meta data), such as Ott et al. (2013) and Li et al. (2014), did not combine this approach with other approaches. In addition to being independent of the system, i.e., it needs only the text, this approach provides quite good results (see Table 9 for examples). However, the major weakness is that it requires a large number of features for being effective. In our previous research (Budhi et al., 2017, 2021), we proved that BOW needs at least 500,000 features to achieve good results with real world datasets such as Yelp!. Similarly, a 1 M vocabulary has been previously used for CBOW (Ren and Ji, 2017). This requirement slows down the training process considerably and usually requires very powerful computing facilities.

The second form of content-based approach attempts to extract information and property of the text, such as the length of text, total words, and total sentences, in addition to linguistic characteristics of the text, such as POS, subjectivity, complexity, diversity, similarity and so on (Rout et al., 2016; Zhang et al., 2016; Wahyuni and Djunaidy, 2016; Heydari et al., 2016; Wang et al., 2016; Hazim et al., 2018). Some approaches extract more interesting features, such as readability scores (Hazim et al., 2018), sentiment polarity score (Akram et al., 2018; Rout et al., 2016; Wahyuni and Djunaidy, 2016; Hazim et al., 2018), the existence of spam terms (Bajaj et al., 2017; Rout et al., 2016; Rathore et al., 2018), and the existence of emoticons, tags and URLs in the text body (Rathore et al., 2018). This type of content-based featuring is more lightweight compared to textual-based featuring – for example, we used only 80 features in this study compared to 500,000 features in our previous study. This approach, too, is independent of the system, as it requires only the text. When it is used alone, however, this approach usually does not provide good results; it can deliver better results only when combined with other featuring methods.

Behaviour-based featuring focuses more on the behaviour of the reviewers rather than their reviews. In this approach, features are extracted from the personal information and behaviour of reviewers, such as their user ID, active tenure, ratings given, date of reviews, frequency of reviews, total reviews, and the types of products reviewed (Barbado et al., 2019; Savage et al., 2015; Yuan et al., 2018; Tang et al., 2020; Li et al., 2015; Kumar et al., 2018; Dong et al., 2020). Some researchers have extended their approach by calculating the honesty, trustworthiness and reliability of the reviewers (Wahyuni and Djunaidy, 2016), checking the IP address, location, cookies of the reviewers (Bajaj et al., 2017; Li et al., 2015), and creating matrixes of features (Yuan et al., 2019), among others. This approach is lightweight, needs only a small number of features and, if designed properly, can deliver good results. However, it is fully dependent on the additional information (meta data) provided by the system. Different systems would produce different sets of features, and therefore, any research output can only be applicable to a particular system and the tested datasets.

Several researchers have also combined behaviour-based featuring with content or textual-based featuring to achieve better results (Bajaj et al., 2017; Rastogi et al., 2020; Martens and Maalej, 2019; Wang et al.,

Table 1
An overview of related work.

Featuring type	Dataset Domain (*)	Algorithm
Textual-based	Hotel (public (Ott et al., 2013)	Positive Unlabelled-Learning (PU-L), Naïve Bayes (NB), Support Vector Machines (SVM) (Hernández Fusilier et al., 2015); NB, SVM, Decision Tree (DT), Random Forests (RF), Gradient-Boosted Trees (GB) (Etaiwi and Naymat, 2017); Deceptive Review Identification by Recurrent Neural Network (DRI-RCNN), SVM, Convolutional Neural Network (CNN), Gated Recurrent Neural Network (GRNN) (Zhang et al., 2018)
	Hotel-Restaurant-Doctor (public (Li et al., 2014)	DRI-RCNN, SVM, CNN, GRNN (Zhang et al., 2018); SVM, CNN, Recurrent Neural Network (RNN), GRNN, Bi-directional average GRNN (B-GRNN) (Ren and Ji, 2017)
	Unlabelled samples	PU-L, NB, SVM (Hernández Fusilier et al., 2015)
	Amazon	Bagging(SVMs, Product Word Composition Classifier (PWCC)) (Sun et al., 2016)
	Hotel-Restaurant	Multinomial NB (MNB), k-Nearest Neighbours (k-NN), DT, RF, Rocchio, SVM, Stochastic Gradient Descent (SGD), Minimum Description Length Text (MDLText), Perceptron (Cardoso et al., 2018)
Textual- and content-based	Hotel-Restaurant-Doctor (public (Li et al., 2014)	Sentence-Weighted Neural Network (SWNN), Long Short-Term Memory (LSTM), SVM (Li et al., 2017)
Textual- and behaviour-based	Yelp [CHI-Split] (Hotel-Restaurant, public (Rayana and Akoglu, 2015)	RESCAL, SVM (Wang et al., 2016); CNN (Wang et al., 2017); CNN, Attribute Enhanced Domain Adaptive (AEDA) (You et al., 2018)
Content- and behaviour-based	Hotel (public (Ott et al., 2013)	DT, SVM, NB, Principal Component Analysis (PCA), k-NN (Rout et al., 2016)
	Yelp (Hotel-Restaurant, public (Mukherjee et al., 2013)	Adaptive Boosting (AB), XGBoost, Generalised Boosted Regression Mode (GBM) Gaussian, GBM Poisson, GBM Bernoulli (Hazim et al., 2018); NB, SVM, DT, RF (Zhang et al., 2016)
	Yelp [CHI, NYC, ZIP] (public (Rayana and Akoglu, 2015)	SpEagle, Light-weight SpEagle (Rayana and Akoglu, 2015); SVM, Logistic Regression (LR), Multilayer Perceptron (MLP), NB Rastogi et al. (Rastogi et al., 2020)
	Facebook	Bayesian Network, Jrip, Decorate, RF, DT, k-NN, LR, SVM (Rathore et al., 2018)
	Reviews from different users	Rule-based Spam review detection Bajaj et al. (Bajaj et al., 2017)
	Amazon	Feature-Centric Model for Review Spam Detection (FMRSD) (Akram et al., 2018); Cosine Similarity, Time-series Spam Reviews Detection (Heydari et al., 2016); Iterative Computation Framework (ICF +), Frequent Pattern Growth (FPGrowth), Jaccard Coefficient (Wahyuni and Djunaidy, 2016);
	Google Playstore Apps	AB, XGBoost, GBM Gaussian, GBM Poisson, GBM Bernoulli (Hazim et al., 2018)
Behaviour-based	Yelp [CHI-Split] (Hotel-Restaurant, public (Rayana and Akoglu, 2015)	Behaviour-feature Generative Adversarial Network (bfGAN) (Tang et al., 2020)
	Yelp (variety, public (Barbado et al., 2019)	LR, DT, RF, Gaussian NB (GNB), AB (Barbado et al., 2019)
	Yelp [CHI, NYC, ZIP] (public (Rayana and Akoglu, 2015)	Hierarchical Fusion Attention Network (HFAN) (Yuan et al., 2019); Target product identification and the *meta*-path feature weight calculation (TM-DRD) (Yuan et al., 2018);
	Apple Store App	RF, DT, MLP, SVM, GNB (Martens and Maalej, 2019)
	Amazon	Autoencoder, Neural Decision Forest (Dong et al., 2020); Detect the proportion of reviews that disagree with the mean rating (Savage et al., 2015)
	Mobile_01	HFAN (Yuan et al., 2019)
	Yelp [CHIOP, NYCOP]	TM-DRD (Yuan et al., 2018)
	Yelp (variety)	LR, k-NN, NB, AB, RF, SVM (Kumar et al., 2018)
	Restaurant (Dianping)	SVM (Li et al., 2015)

(*) References are only provided for publicly available datasets.

2017, 2016; Akram et al., 2018; Rayana and Akoglu, 2015; You et al., 2018; Rout et al., 2016; Zhang et al., 2016; Wahyuni and Djunaidy, 2016; Heydari et al., 2016; Hazim et al., 2018; Rathore et al., 2018). Our proposed method takes a unique approach and incorporates behaviour-based features from the products' perspective, which helps improve the fake and genuine class accuracies further after the sampling process. In this study, our behaviour-based features have been designed to be as general as possible – using only four (4) additional pieces of information that usually exist in any online product review system. These are: User ID of the reviewer, Product ID of the reviewed product/service, the review date, and the ranking/stars given to the product/service in the review. We process the above-mentioned information to extract 49 behaviour-based features. We also deal with the imbalance issue, which is known to be an impediment for training machine learning detection of minority samples from majority samples and could eventually lead to false sense of success. To do so, we propose a dynamic random sampling method that can increase the minority class or decrease the majority class based on their current composition immediately before the start of the training process. We implement this approach to a wide range of machine learning and deep learning classifiers that have performed well in our previous studies. In addition, to speed up the slow process of experiments using *n*-fold CV, we propose a parallel version of our approach to significantly accelerate the process in a parallel environment, i.e. in a High-Performance Computing (HPC) facility.

3. Datasets and features

As discussed in the previous section, we have used Yelp fake review datasets from Rayana and Akoglu (Rayana and Akoglu, 2015). There is a total of four datasets, namely YelpChi Hotel, YelpChi Restaurant, YelpNYC and YelpZIP. These datasets are highly imbalanced, since only around 10% of total reviews are fake (see Table 2. They contain records that have been manually labelled from the original Yelp! dataset; however, because of information reduction, they are not as complete as the raw dataset published by Yelp! (Yelp, 2019). Information inside each Yelp fake review dataset includes the user ID, product ID, given ratings, date, fake/genuine label and user review text. The YelpChi datasets contain reviews for a set of restaurants and hotels in the Chicago area; the YelpNYC dataset contains reviews for restaurants located in New York City; and the YelpZIP dataset contains reviews for restaurants with various zip codes in the United States covering a geographical region that includes New York, New Jersey, Vermont, Connecticut, and Pennsylvania.

We defined our features based on the information provided in the

Table 2
The statistics of Yelp fake review datasets.

Name	Total sample	Fake reviews		Genuine Reviews		Total users	Total products
		Total	%	Total	%		
YelpChi H (Hotel)	5854	778	13.29	5076	86.71	5026	72
YelpChi R (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

datasets, and they are listed in Table 3. Next, we built a feature-extraction mechanism to be used in supervised machine learning classification training. Of the features, 80 are content-based features processed directly from the text; four are additional information about the review (user ID, product ID, review date and the rating given); and 25 features are behaviour-based features extracted from the former 4 features. Additionally, we added another 24 behaviour-based features extracted from the products or services that were reviewed. Several content-based features were extracted using several components from the Natural Language Toolkit (NLTK, 2019) and textstat (Bansal and Aggarwal, 2019). Similarly, the remaining features, such as detecting words with capitalised 1st letter, negative terms and elongated words, creating all linguistic characteristic features, all user-perspective and product-perspective behaviour-based features, were extracted using functions and formulas that we built using the Python programming language.

The reason for creating many features is to provide as many traits of the user-review data as possible to train machine learning. A large number of facet features will help the machine learning predictors generalise and recognise target classes. The feature extraction process is designed to have four separate sections (see Fig. 1). The first section is content-based featuring that extracts features directly from the text reviews. Therefore, we call these features *review-perspective content-based* features. The 80 features in this section are categorised into 6 groups based on their functions (see Table 3 for the justification of each function). Group A is about some basic information that we can extract from the text. Group B (POS) consists of 36 POS tags based on Penn POS (Buchholz, 2002). In Group C, we capture the linguistic traits of the text. Group D (readability scores) consists of features extracted using functions from the textstat project (Bansal and Aggarwal, 2019). We compiled the spam dictionary from various Internet resources (Shuteyev, 2018; Perelsztejn, 2017; Pels, 2019) for Group E (spam terms), whereas we used SentiWordNet 3.0 (Baccianella et al., 2010) as a dictionary to extract sentiment features for Group F (sentiment analysis).

The second section is about the information or meta data that supports the reviews, and it is further split into two groups. The first group (G) includes information about the reviewer: their ID, date of birth, location, city, IP address, and so on. The second group (H) includes additional information about the review itself, such as the reviewed product/service, rank/star given to the product/service by the reviewer, and the review date. While many features could exist for Group G, we use only one feature for this group in our study: User ID of the reviewer, since it is the only attribute in Rayana & Akoglu's datasets for this group.

The final two sections are behaviour-based featuring sections. We call Section 3 *user-perspective behaviour-based* features because features from the perspective of the user are created in this section. The section consists of 25 reviews divided into 3 groups: basic user behaviour, user behaviour based on time difference, and their behaviour based on the ranks/stars given by them. All features in this section will be the same for all reviews by the same user (grouped by the user ID). The final section is still about behaviour of the reviewers and is similar to Section 3, but from the perspective of the reviewed product/service; therefore, we call this section *product-perspective behaviour-based* features. All features in this section will be the same for all reviews about the same product (grouped by the product ID). The features from the products' perspective are expected to improve detection accuracy of the genuine class. While some features have already been suggested in the literature, many of these features are new to the literature and help to boost the accuracy of fake review class detection.

4. Methods

The datasets used in our study are imbalanced datasets, which could affect prediction (Zhang et al., 2016; Mukherjee et al., 2013; Hu et al., 2019). Hence, our sampling process has been designed to optionally implement random over or under-sampling in order to overcome the issue of imbalanced data. The flowchart for the sampling process and the entire system can be seen in Figs. 2 and 3, respectively.

Once the fake review dataset has been loaded, all required settings, such as the feature groups used, type of sampling (under-sampling, over-sampling or no sampling), number of folds for the *n*-fold CV, are loaded to properly drive training and testing. All extracted features are normalised using the Min-Max Normalisation technique. Without normalisation, the scale range of each feature could be different, which will impact the training process. After normalisation, the data is split to be *n*-fold and then grouped into either the training set or the test set for each fold.

Next, the sampling process is applied to the training sets. For random under-sampling or over-sampling, we use methods similar to those implemented by Hu et al. (2019); these methods have shown good potential to overcome the class imbalance problem (Budhi et al., 2021). Note that we apply the sampling methods to only the training sets. This helps avoid the possibility of overfitting during the training phase, especially on the over-sampling method where a new record is created by randomly copying one of the existing records. The sampling process works dynamically based on the current composition of minority and majority class features immediately before the training process begins. This will reduce the majority class in under-sampling, or increase the minority class in over-sampling, to a new ratio that can be set beforehand. However, for the sake of simplicity, we always set the ratio to be 1:1 in this study. Following sampling, as depicted in Fig. 3, the *n*-fold CV process is run according to the assigned classifiers. Finally, after determining the best classifier for detecting fake reviews, all information and the detailed results are written to a file.

A parallel processing approach is implemented to accelerate the *n*-fold CV process (see Fig. 4). The idea is simple; all CV processes are assigned to different CPUs so that, instead of iterating for *n* times, the CV processes run on *n* parallel CPUs. Upon completion, each CV fold process writes its result to a temporary file and checks whether other processes have completed. If other processes have completed, this process compiles all the temporary results from other CV processes, calculates the average of measurements and the running time of the entire process, and finally compiles the results file before terminating. Otherwise, the process does no further work and terminates. The CV process has been split into two different pieces of code. The first piece is related to initialisation and data preparation, which, upon completion, will create *n* jobs of the second piece for running each CV fold process on different CPUs. See Fig. 4 for the list of procedures that are handled by each piece of code.

We have implemented and tested a number of machine learning and deep learning classifiers for detecting fake reviews in this work. These classifiers are often used in text analysis in general, and they have shown excellent performance in our previous research on textual-based

Table 3
Features.

No.	Group	Description	References	Justification
Section 1: Review-perspective content-based features				
1–4	A: Basic text information	Total (letters, words, stop words, sentences) in the review	(Akram et al., 2018; Rout et al., 2016; Zhang et al., 2016; Martens and Maalej, 2019)	Fake reviews are written to falsely persuade people about something and various means are used to achieve this. In this group, we capture the characteristics of the text to differentiate between fake and genuine reviewers
5		Total words with capitalised 1st letter	(Rout et al., 2016)	
6		Total negative terms (e.g. 'does not', 'do not', 'will not', etc.)	(Rout et al., 2016)	
7		Total elongated words (e.g. 'Yesss', 'fiiine', 'yoouu', etc.)	–	
8–9		Total exclamation and question sentences	–	
10		The existence of weblink inside the text	(Rathore et al., 2018)	
11–46	B: POS	Total existence of 36 Tags of Penn POS (Buchholz, 2002)	(Rout et al., 2016; Zhang et al., 2016; Wahyuni and Djunaidy, 2016)	As in Group A, here, we capture text traits based on Penn POS, such as verbs, nouns, and adjectives.
47	C: Linguistic characteristics	The ratio of adjectives and adverbs	(Zhang et al., 2016)	In this group, we define more complex/advanced traits of the text by calculating the ratio of adjectives to verbs, averaging total words per sentence, etc. This is because the writing of professional fake reviewers is usually more structured than common reviewers.
48		Average of number of words per sentence	(Rout et al., 2016; Zhang et al., 2016)	
49		The ratio of word repetition to total words	(Zhang et al., 2016)	
50		The average number of letters per word	(Rout et al., 2016; Hazim et al., 2018)	
51		Average of words with 1st capital to total sentences.	–	
52		The ratio of words with 1st capital to total words	(Zhang et al., 2016)	
53–55		Total of (1st, 2nd, 3rd) person pronouns	(Akram et al., 2018)	
56–58		The ratio of (1st, 2nd, 3rd) person pronouns to total pronouns	(Luca and Zervas, 2016; Rastogi et al., 2020; Zhang et al., 2016)	
59–65	D: Readability scores (Bansal and Aggarwal, 2019)	Flesch reading ease, Simple Measure of Gobbledygook (SMOG) index, Flesch Kincaid grade, Coleman-Liau index, Gunning fog index, Dale–Chall readability and Linsear Write formula.	–	Here, we score the readability of the text with the purpose of differentiating fake reviews which try to convince the reader about something to genuine reviews that are written to express the reviewer's like or dislike of a product/service.
66		Automated readability index (ARI)	(Hazim et al., 2018)	
67		Difficult words	–	
68		Estimation of school grade level required to understand the text.	–	
69	E: Spam term	Total spam terms	(Bajaj et al., 2017; Rout et al., 2016; Heydari et al., 2016; Rathore et al., 2018)	Fake reviewers have tendency to use jargon and bombastic terms that can be categorised as spam terms.
70		Average of Spam term per sentence	–	
71		The ratio of spam term to non-spam words	–	
72	F: Sentiment analysis (Baccianella et al., 2010)	Total of sentiment terms	–	While sentiment words are used by both fake and genuine reviewers, fake reviewers could use them more often in order to convince readers about liking/disliking a product/service.
73–75		Total of (positive, neutral, negative) of sentiment terms	(Akram et al., 2018; Wahyuni and Djunaidy, 2016)	
76–77		The ratio of (positive, negative) sentiment to neutral terms	(Zhang et al., 2016)	
78		The ratio of negative to positive sentiment terms	(Rastogi et al., 2020; Zhang et al., 2016)	
79–80		(Positive, negative) sentiment scores	(Akram et al., 2018; Rout et al., 2016; Hazim et al., 2018)	
***Section 2*: Basic Information**				
81	G: User Info	User ID	(Bajaj et al., 2017; Barbado et al., 2019)	User ID or/and other information that can point to someone, a place, or computer IP, is useful to identify a fake reviewer based on the previous evidence.
82–84	H: Basic info	Product ID, rank/star given, the month when the review was written	(Rout et al., 2016; Zhang et al., 2016; Hazim et al., 2018; Heydari et al., 2016)	Here, we capture basic habits of the reviewer.
***Section 3*: User-perspective behaviour-based features**				
85–88	I: Basic user behaviour	User tenure, total reviews given by the user, total products reviewed by user and total rank/stars given by the user.	(Barbado et al., 2019; Akram et al., 2018; Heydari et al., 2016; Kumar et al., 2018; Dong et al., 2020)	In this group, we feature the basic behaviour of users.
89–93	J: Behaviours based on time difference	Minimum, maximum, mean, median and coefficient of variation of the time difference between two consecutive reviews.	(Kumar et al., 2018; Heydari et al., 2016)	These features capture details of user's behaviour based on the time they are active.
94–100	K: Behaviours based on rating/star given	Minimum, maximum, mean, mode, variance, standard deviation, and entropy of ratings/stars given by the reviewer	(Savage et al., 2015; Rout et al., 2016; Kumar et al., 2018; Dong et al., 2020)	We create these features to present more behaviour of users based on their tendency to give certain stars/ranks in their reviews.
101–105		Total of 1, 2, 3, 4 and 5 stars given by the reviewer	(Barbado et al., 2019; Dong et al., 2020)	
106–108		The ratio of the number (positive, neutral, negative) rank/stars given to total rank/stars given by the reviewer	(Barbado et al., 2019; Zhang et al., 2016; Dong et al., 2020)	

(*continued on next page*)

Table 3 (*continued*)

No.	Group	Description	References	Justification
109		Ratio of the number of positive to negative rank/ stars given	(Barbado et al., 2019; Zhang et al., 2016; Dong et al., 2020)	
***Section 4*: Product-perspective behaviour-based features**				
110–112	L: Basic user behaviour	Total reviews for the product/service, total users who reviewed the product/service and total rank/ stars given for the product/service.	–	In this group, we feature the basic behaviour of users in relation to a particular product.
113–117	M: Behaviours based on the time difference	Minimum, maximum, mean, median and coefficient of variation of the time difference between two consecutive reviews of the product/service.	–	These features capture details of user's behaviour based on the time certain products/ services were reviewed.
118–124	N: Behaviours based on rating/star given	Minimum, maximum, mean, mode, variance, standard deviation, and entropy of rating/stars given to the product/service.	–	These features capture the behaviour of users towards a product based on the stars/ ranks given to a product.
125–129		Total of 1, 2, 3, 4, and 5 stars given to the product/ service.	–	
130–132		The ratio of the number (positive, neutral, negative) rank/stars given to the total rank/stars given to the product/service.	–	
133		The ratio of the number positive to negative rank/ stars given	–	

sentiment analysis (Budhi et al., 2017, 2021; Lo et al., 2017), classification and ranking of high value audiences (Lo et al., 2015, 2016) and malicious web domain identification (Hu et al., 2019; Hu et al., July, 2016). Specifically:

a. Four single classifier models LR (Menard, 2010), Linear-kernel SVM (Campbell and Ying, 2011; Chang and Lin, 2011), MLP and DT are considered. An improved version of the MLP (Glorot and Bengio, 2010; Kingma and Adam, 2015), which is more reliable than the base version (Rumelhart et al., 1986), is used in this study. The DT model (Quinlan, 1986) was included because it is typically used as a base classifier for ensemble models (e.g., Bagging Predictors (BP), RF, and AB).
b. Four ensemble models RF (Breiman, 2001), GB (Friedman, 2001), BP (Breiman, 1996) and AB (Zhu et al., 2009) are used. In addition to the default base predictor (DT), three other single classifiers (LR, SVM, and MLP) are applied as the base predictors for BP, and the LR and SVM are used as base predictors for AB. The MLP is not compatible with AB.
c. Two deep learning models CNN (Yu et al., 2016; Krizhevsky et al., 2017) and Feed-Forward Deep Learning (FFDL) are used. The CNN has been successfully used in many areas, including fake review detection (Zhang et al., 2018; Budhi et al., 2021). Theoretically, the training performance of CNN is only slightly worse than the standard feed-forward neural network (Krizhevsky et al., 2017). FFDL is an MLP built using deep learning libraries such as TensorFlow, Theano and Keras (Lee et al., 2017). While similar to the MLP, it also inherits some advantages of deep learning components, such as GPU-enabled processing, more efficient modelling of neural layers and several choices of activation functions.

We built all machine learning classifiers and ensembles using scikit-learn components (Scikit-learn, 2019). To ensure the results can only be affected by the implementation of our approach and not by the modification of classifier parameters, we used the default parameters for all single classifiers. We applied the same approach of using the default parameters for the ensemble models, with the exception of applying different base-classifiers for BP and AB, in order to investigate the impact of different base classifiers for fake review detection. For deep learning models, the classifiers were built using Keras components (Keras, 2019). Since Keras does not provide default settings for the CNN and FFDL, we used previously developed models that performed well in our previous study (Budhi et al., 2021), with minor adjustments for tackling fake review detection. The configurations of deep learning models used in this study can be seen in Table 4.

5. Measurements

We implemented measurement components from scikit-learn (Scikit-learn, 2019) for performance measurements in our experiments. These measurements and their respective formulas can be found in Table 5.

6. Results and discussion

In this work, all experiments were conducted using the 10-fold CV method, except for parallel CV experiments, where we investigated the performance of parallel processing for CV. We used components from scikit-learn (Scikit-learn, 2019) for the machine learning classifiers, Min-Max normalisation formula, CV process, and measuring the results (accuracy, precision, recall and F1). Additionally, we used components from the Natural Language Toolkit (NLTK, 2019) and textstat (Bansal and Aggarwal, 2019) for extracting the features. As discussed earlier, we used Yelp's fake review datasets from Rayana and Akoglu (Rayana and Akoglu, 2015). The experiments focused on the following:

1. investigating whether random over-sampling or under-sampling can increase the accuracy of the minority class (fake reviews).
2. investigating the effect of content-based and behaviour-based features.
3. identifying the best machine learning or deep learning classifier to detect fake reviews.
4. investigating the improvement in processing speeds with parallel processing in the *n*-fold CV process.

6.1. Sampling or not sampling

As briefly mentioned earlier, imbalanced data can, theoretically, affect the prediction performance. To investigate this, we conducted a set of experiments on the four datasets from Table 2 and all the features in Table 3, using three best single classifiers identified in our previous studies (Budhi et al., 2017, 2021), namely the MLP, LR and SVM linear kernel. The results of these experiments are listed in Table 6. We calculated the overall accuracy using sA. For detailed accuracy, we calculated the sA of only the particular class, i.e., total correct predictions of fake class testing samples against total fake class testing

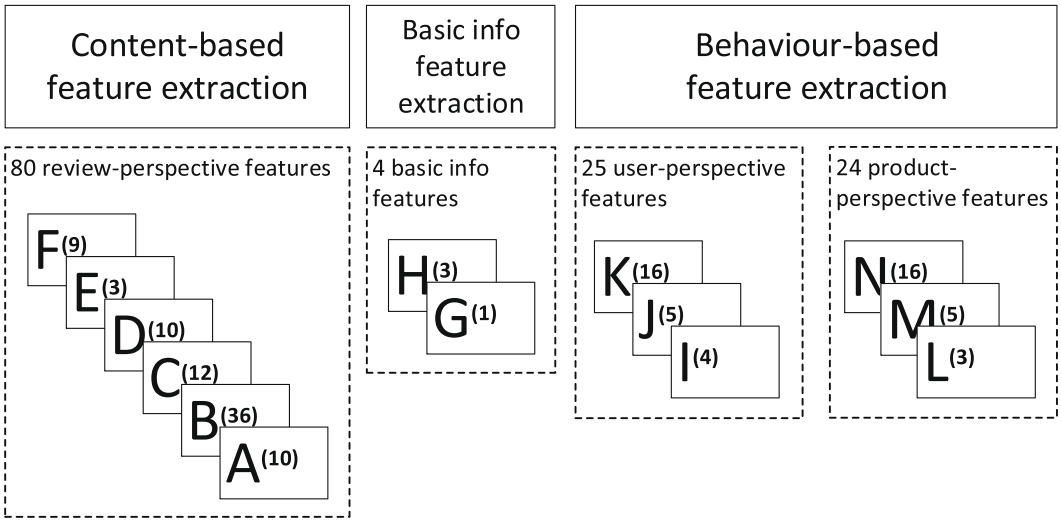

Fig. 1. Design of the input feature extraction process.

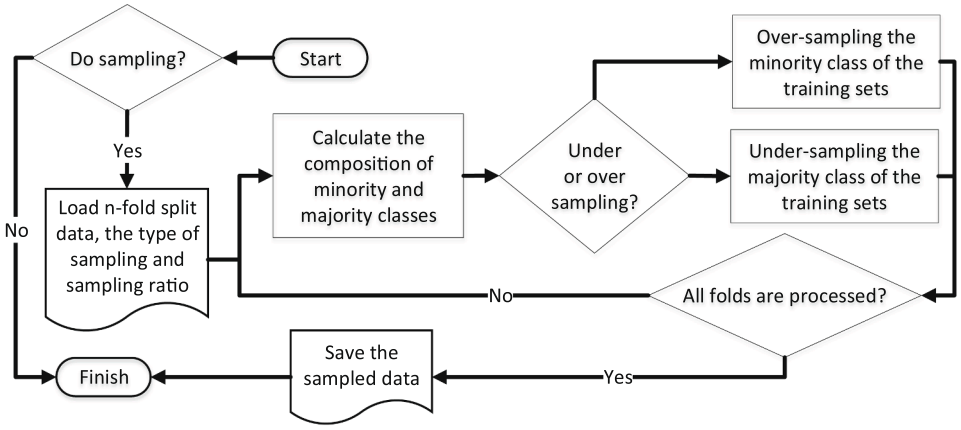

Fig. 2. The sampling process.

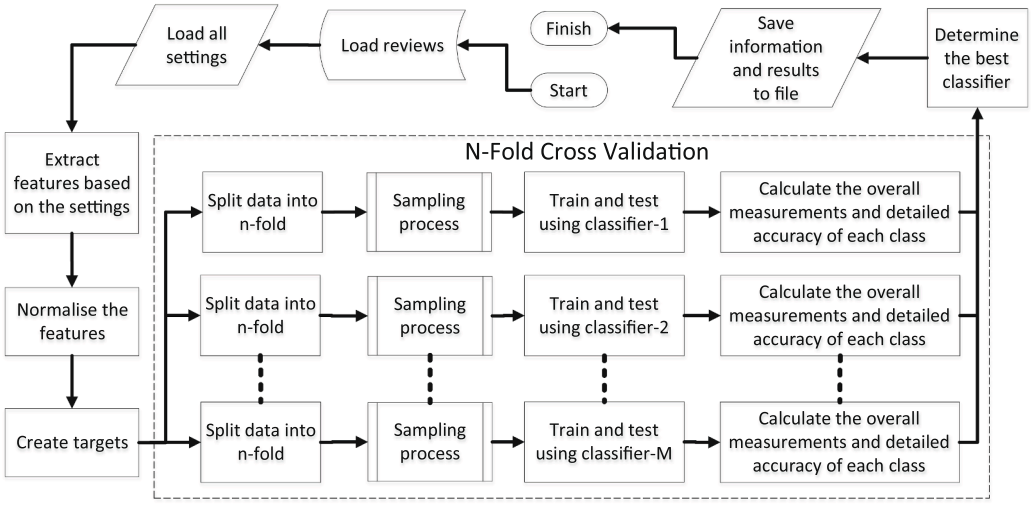

Fig. 3. Overall design.

samples. It needs to be mentioned that since fake review detection is a binary-target problem, detailed sA and binary recall (bR) will produce the same score for each class.

Results in Table 6 show that overall measurements of the prediction are good for all datasets, with best performance for the smaller datasets (almost 100% for YelpChi Restaurant and 97% for YelpChi Hotel datasets). However, upon closer inspection of the accuracy of each class, we observe that the accuracy of minority class (i.e., fake reviews) in larger

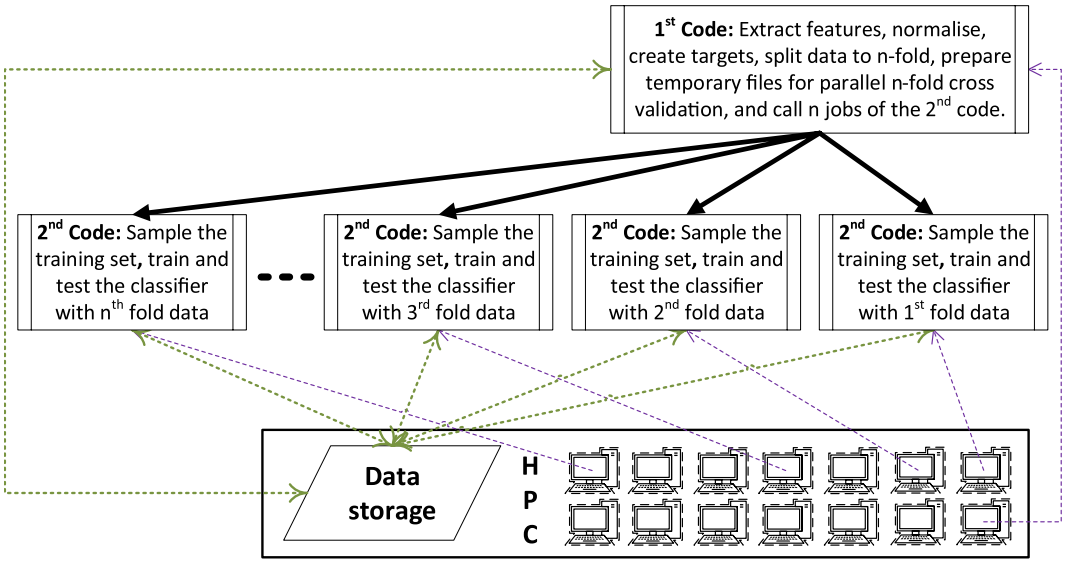

Fig. 4. Design of the parallel n-fold CV process.

datasets (YelpZIP and YelpNYC) is much lower compared to the majority class (i.e., genuine reviews). This is undesirable because we are building a fake review detection system, not the opposite. The accuracy comparison also shows that measuring only the overall results, without checking each target class, could lead to incorrect assumptions about the effectiveness of the algorithms. To overcome this problem, we implemented random sampling techniques.

Random over-sampling and under-sampling have their strengths and weaknesses. The main strength of over-sampling is it provides enough data for the minority class, which is essential for training in machine learning. However, random over-sampling creates duplicates, which can lead to overfitting. In contrast, random under-sampling does not create duplicates, since it reduces the size of the majority class. However, under-sampling can lead to deletion of some important traits of the majority class. The other weakness of under-sampling is that, if the minority class is too small, it will significantly reduce the number of majority class samples, when theoretically, machine learning algorithms need a large amount of data to perform well. Since the imbalance problem is observed only on the two big datasets (YelpNYC dan YelpZIP) we focused on these two datasets for our experiments on data sampling. The results of these experiments can be seen in Figs. 5 and 6.

Fig. 5 shows that, for both datasets, random sampling methods reduced the overall accuracy and recall but increased the overall precision. Most importantly, from Fig. 6, we can see that under and over-sampling increased the accuracy of fake class from 16% to 86% and 85% for the MLP, respectively, on the YelpZIP dataset. However, the genuine class accuracy was reduced from 98% to 68% and 69%. Similar results were obtained for the YelpZIP and YelpNYC datasets on all three classifiers tested. These results prove that sampling methods can improve the detection of fake reviews on imbalanced datasets but, at the same time, increase the misdetection of genuine reviews.

Table 4
Configurations of deep learning classifiers.

Model	Configuration
FFDL Base	3 × Dense(100, relu) - Dense(2, sigmoid)
CNN Type 1	3 × Convolution2D(32, relu, kernel 3x3) - MaxPooling – 3 × Convolution2D (64, relu, kernel 3x3) – MaxPooling - Dense(1024, relu) - Dense(512, relu) - Dense(2, sigmoid)
CNN Type 2	2 × Convolution2D (32, relu, kernel 3x3) - MaxPooling – 2 × Convolution2D (64, relu, kernel 3x3) - MaxPooling – 2 × Convolution2D (128, relu, kernel 3x3) - MaxPooling - Dense(1024, relu) - Dense(512, relu) – Dense(2, softmax)

6.2. Effect of features

In this section, we present the results of our investigation into the effect of each feature group (listed in Table 3) on the prediction performance. 10-fold CV experiments were run on the MLP classifier, which is the best classifier from previous experiments, and, for the sake of simplicity, only the biggest dataset (YelpZIP) was used. Features were built using under-sampling and over-sampling, which can improve the performance of fake review detection on the YelpZIP dataset as discussed in the previous section.

In Fig. 7, we can see that Groups I, J and K (user-perspective behaviour-based features), in general, improved the prediction accuracy the most compared to other groups. In contrast, Groups A-F (review-perspective content-based features) and G-H (basic information extracted directly from the dataset structure), while still significantly improving the detection of fake review class, provided smaller improvements to the overall and detailed accuracy. There were some interesting observations for Groups L, M, and N (product-perspective behaviour-based features). These groups did not increase the detection accuracy of fake review class as much as other groups, but also did not reduce the detection accuracy of genuine class as much as other groups. However, when combined with other groups, the unique attributes of these groups helped increase the detection accuracy of genuine review class, which is otherwise heavily reduced due to sampling. Fig. 8 and Table 7 show that all feature groups, when combined, can improve detection accuracy of fake review class without making great reduction in the detection accuracy of genuine review class. Regarding the smaller datasets in particular, the overall group combination can provide excellent accuracies for both fake and genuine review classes (see Table 6 and Fig. 9). While some behaviour-based combination such as IJK or IJKLMN provided the highest accuracies for the fake class in over and under-sampling, their accuracies in terms of the genuine class were not good. Hence, we consider the combination of all groups the best, since such a combination provided better scores than other combinations. Therefore, all the features were used for the next set of experiments.

6.3. Investigation of best classifiers

We ran experiments with several machine learning, deep learning

Table 5
Measurement functions and formulas.

No	Name	Sklearn Function	Equation
1	Exact-match/ subset-accuracy (sA)	accuracy_score()	$sA = A(y,\hat{y}) = \frac{1}{n_{samples}}\sum_{k=0}^{n_{samples}-1} 1(\hat{y_k} = y_k)$ where y is the set of predicted pairs, $\hat{y}$ is the set of true pairs, and $n_{samples}$ is total samples.
2	Balanced-accuracy (blA)	balanced_accuracy_score()	$blA = \frac{1}{2}\left(\frac{tp}{tp+fn} + \frac{tn}{tn+fp}\right)$ where tp is true positive, fn is false negative, tn is true negative and fp is false positive
3	Weighted-precision (wP) and binary-precision (bP)	precision_score()	$wP = \frac{1}{\sum_{l\in L}\lvert\hat{y}_l\rvert}\sum_{l\in L}\lvert\hat{y}_l\rvert P(y_l,\hat{y}_l) bP = P(y_l,\hat{y}_l) = \frac{tp}{tp+fp}$ where L is set of classes, y_l is the subset of y with class l
4	Weighted-recall (wR) and binary-recall (bR)	recall_score()	$wR = \frac{1}{\sum_{l\in L}\lvert\hat{y}_l\rvert}\sum_{l\in L}\lvert\hat{y}_l\rvert R(y_l,\hat{y}_l) bR = R(y_l,\hat{y}_l) = \frac{tp}{tp+fn}$
5	Weighted-Fmeasure (wF1) and binary- Fmeasure (bF1)	f1_score()	$wF1 = \frac{1}{\sum_{l\in L}\lvert\hat{y}_l\rvert}\sum_{l\in L}\lvert\hat{y}_l\rvert F1(y_l,\hat{y}_l) bF1 = F1(y_l,\hat{y}_l) = 2*\frac{P(y_l,\hat{y}_l)*R(y_l,\hat{y}_l)}{P(y_l,\hat{y}_l)+R(y_l,\hat{y}_l)}$
6	Average-precision (AP)	average_precision _score()	$AP = \sum_n (R_n - R_{n-1}) P_n$ where P_n and R_n are the precision and recall at the n-th threshold
7	Area under the receiver operating characteristic curve (AUC)	roc_auc_score()	$AUC = \frac{2}{c(c-1)}\sum_{j=1}^{c}\sum_{k>j}^{c} p(j\cup k)(AUC(j\lvert k) + AUC(k\lvert j))$ where c is the number of classes and $AUC(j\lvert k)$ is the AUC with class j as the positive class and class k as the negative class

Table 6
Results of prediction without feature-sampling the datasets

Classifier	Dataset	Overall Measurements				Detailed sA (%)	
		sA (%)	wP (%)	wR (%)	wF1 (%)	Fake	Genuine
MLP	YelpZIP	87.37	84.55	87.37	84.31	17.58	97.99
	YelpNYC	89.92	87.00	89.92	86.33	7.73	99.30
	YelpChi R	99.70	99.70	99.70	99.70	98.27	99.92
	YelpChi H	97.83	97.84	97.83	97.77	86.42	99.61
	Average	**93.70**	**92.27**	**93.70**	**92.03**	**52.50**	**99.20**
LR	YelpZIP	86.99	83.33	86.99	82.54	8.04	99.00
	YelpNYC	89.76	86.00	89.76	85.23	1.64	99.83
	YelpChi R	99.73	99.73	99.73	99.73	98.10	99.98
	YelpChi H	96.24	96.22	96.24	96.05	75.49	99.41
	Average	**93.18**	**91.32**	**93.18**	**90.88**	**45.82**	**99.55**
SVM	YelpZIP	86.96	83.48	86.96	81.75	4.25	99.55
	YelpNYC	89.75	85.44	89.75	84.92	0.08	99.99
	YelpChi R	99.74	99.74	99.74	99.74	98.20	99.98
	YelpChi H	97.69	97.68	97.69	97.63	85.52	99.57
	Average	**93.54**	**91.59**	**93.54**	**91.01**	**47.01**	**99.77**

and ensemble models that are often used for text analysis to determine the best classifier for fake review detection. The selection of classifiers was inspired by the excellent results they provided in our previous studies (Budhi et al., 2017, 2021). No sampling settings were used for the two small datasets (YelpChi Hotel and Restaurant), whereas under and over-sampling of features were conducted for the larger datasets (YelpNYC and YelpZIP). All the features listed in Table 3 were used because, as discussed above, we consider the combination of all features to be the best for this problem.

We can see in Fig. 9 and Fig. 10 that all the classifiers used for prediction in our experiments performed well for both the smaller datasets, especially the YelpChi Restaurant dataset. Similarly, after applying

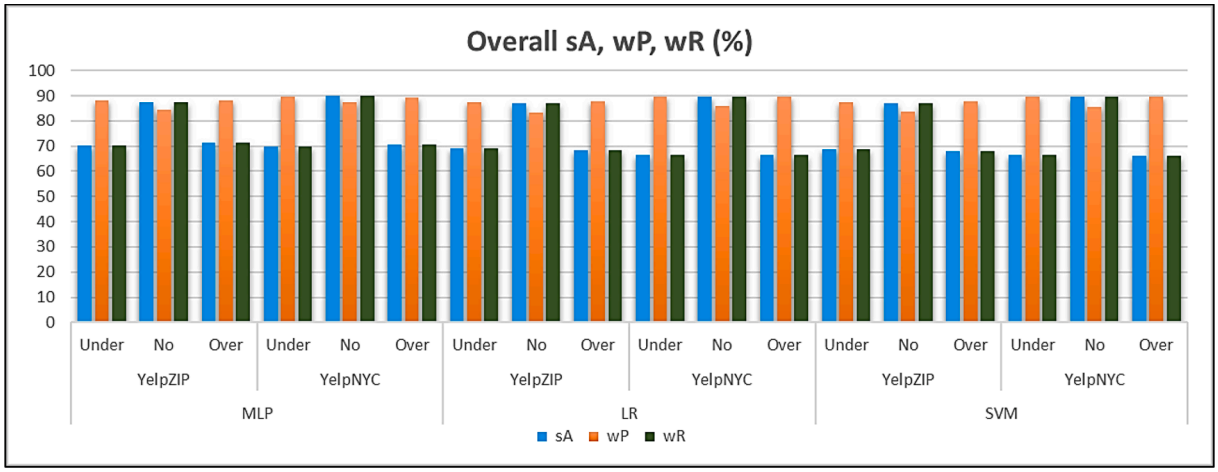

Fig. 5. Overall sA, wP, and wR of YelpZIP and YelpNYC.

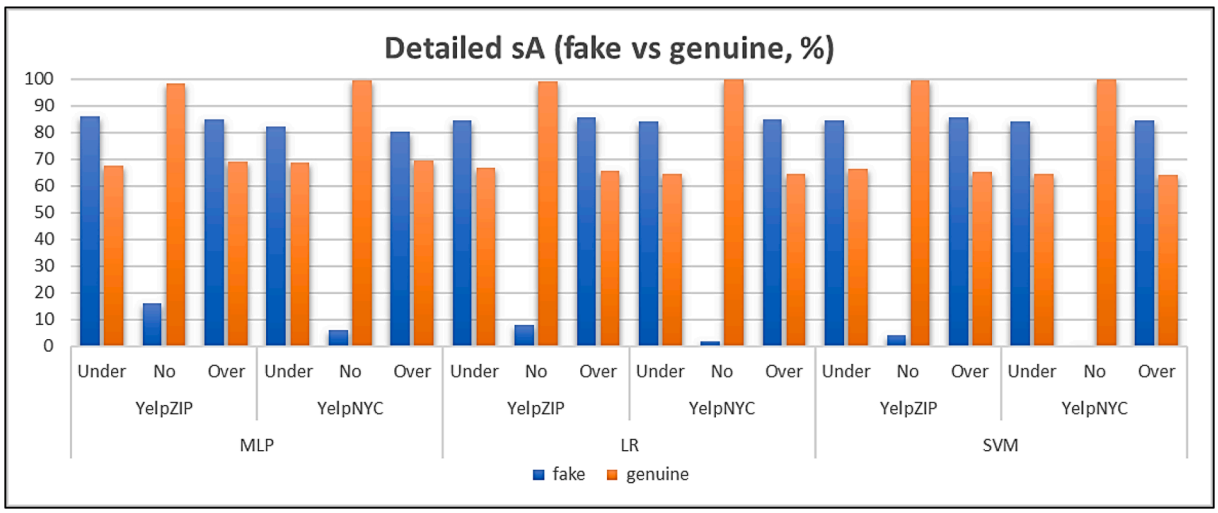

Fig. 6. Detailed sA for each target class (Under, No, and Over-sampling).

under-sampling on the big datasets (see Figs. 11 and 12), the accuracy of classifiers for the fake review class was quite high. The prediction accuracy of all classifiers, except the DT and AB(LR), was above 80%, with accuracies of DT and AB(LR) being 70% and 79%, respectively. The highest subset accuracy for the fake review class was obtained by the CNN Type 1 with over-sampling – above 88% for both the bigger datasets (YelpNYC and YelpZIP). However, with sampling, the prediction accuracies for the genuine review class were not as high (only between 65% and 67%). Nevertheless, these results are acceptable since the purpose of this research is to detect fake reviews, and not genuine reviews. The second-best classifier was the GB ensemble with 87% accuracy for the fake review class.

We also discovered that several classifiers, such as the DT, RF, BP (DT) and CNN Type 2 did not perform well with over-sampling (see Figs. 13 and 14). While these classifiers increased the accuracy of the fake review class, the accuracy was not as high as other classifiers. In the case of CNN, we can see that CNN Type 1, which uses 3 convolutional layers of 32 neurons followed by 3 more convolutional layers of 64 neurons, is more stable than CNN Type 2. CNN Type 2, which uses 2×32 convolutional layers, 2×64 convolutional layers and 2×128 convolutional layers, shows more unstable behaviour. Whilst it provided slightly better results than Type 1 with under-sampling, it performed poorly with over-sampling, especially on the biggest dataset (YelpZIP).

The DT is the likely source of the problem, since it is the base classifier of both the RF and BP(DT). The DT with default parameters cannot be applied to big datasets. Under-sampling reduces the size of the training set to almost half the smaller class (fake class), which is only 10% in YelpNYC and 13% in YelpZIP. In this case, the DT and DT-based ensemble models can run well. On the other hand, over-sampling increases the amount of training data to be almost double the bigger class (genuine class). On 1:1 ratio oversampling, the process randomly replicates the fake class samples to make them as numerous as the genuine class samples. This process made the over-sampled YelpNYC and YelpZIP too big to be handled well by the DT and DT-based ensemble models. However, when combined with a boosting procedure such as the AB ensemble, the DT provides better accuracy for detecting the fake review class.

6.4. *Speeding up CV using parallel processing*

N-fold CV often takes time, especially when we choose higher *n*, such as the 10-fold CV. In a 10-fold CV process, we need to run training and testing 10 times with different sets of train-test data. If run only for a single set of features, a single dataset on a single classifier, its processing time is still reasonable. However, the time required is significantly higher when investigating combinations of several sets of features from multiple datasets and with multiple classifiers.

Our experiments were run on the HPC facility at the University of Newcastle (UoN), Australia. This facility has 4,000 usable cores for 120 CPU nodes and 6 GPU nodes, with up to 512 GB RAM. To further speed up the experiments, each validation process in CV was individually assigned to different CPU nodes, as depicted in Fig. 4; and all processes should, theoretically, run at the same time. To investigate the impact of this method on accelerating the overall process, we ran CV for settings ranging from 2- to 10-fold, using all feature groups, the biggest dataset (YelpZIP), LR classifier, and 10 experimental runs for each CV setting. The experiments were run in a cluster of UoN HPC with 32 CPU nodes. They were also run in a normal (not dedicated) environment of HPC, where jobs from various projects simultaneously using the HPC are queued together and wait in a queue for a free CPU node.

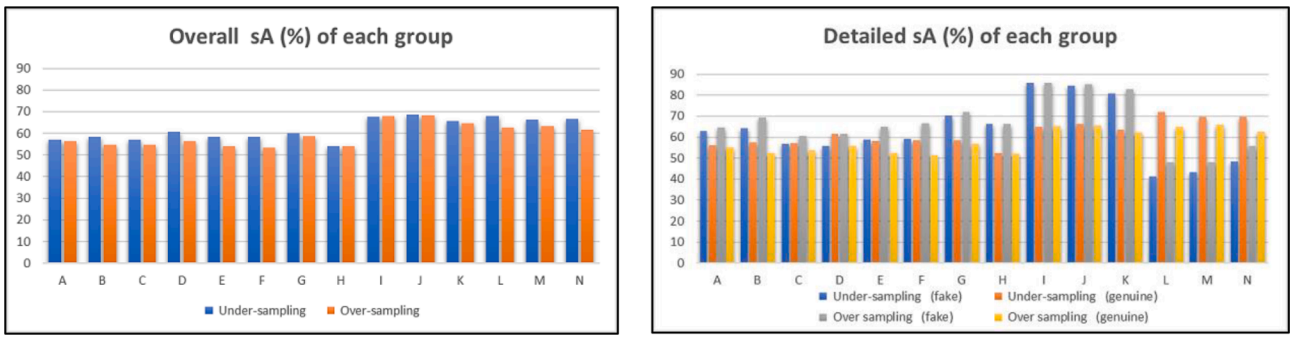

Fig. 7. Overall and detailed sA for each features group (MLP, YelpZIP).

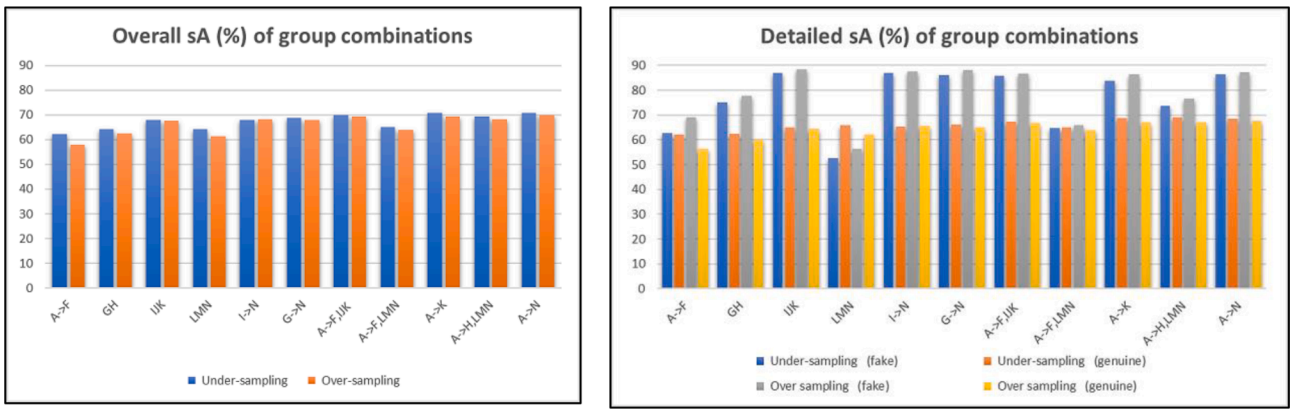

Fig. 8. Overall and detailed sA of features groups combinations (MLP, YelpZIP).

Table 7
Overall and detailed sA of feature group combinations (MLP, YelpZIP)

Content-based features (Review-perspective)						Basic Info		Behaviour-based features						Under-sampling (sA, %)			Over-sampling (sA, %)		
								User-pers			Product-pers								
A	B	C	D	E	F	G	H	I	J	K	L	M	N	Fa	Ge	Ov	Fa	Ge	Ov
✓	✓	✓	✓	✓	✓									62.81	62.17	62.25	68.96	56.34	58.02
						✓	✓							75.07	62.49	64.15	77.65	60.20	62.50
								✓	✓	✓				86.82	65.08	67.95	**88.46**	64.41	67.58
											✓	✓	✓	52.57	66.00	64.23	56.45	62.15	61.40
								✓	✓	✓	✓	✓	✓	**87.00**	65.23	68.11	87.51	65.47	68.38
						✓	✓	✓	✓	✓	✓	✓	✓	86.05	66.13	68.76	87.99	64.97	68.01
✓	✓	✓	✓	✓	✓			✓	✓	✓				85.73	67.46	69.87	86.74	66.74	69.38
✓	✓	✓	✓	✓	✓						✓	✓	✓	64.61	65.12	65.05	66.02	63.79	64.09
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				83.72	68.80	**70.92**	86.23	66.91	69.46
✓	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓	73.66	**68.97**	69.45	76.49	67.04	68.28
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	86.42	68.40	70.78	87.33	**67.47**	**70.09**

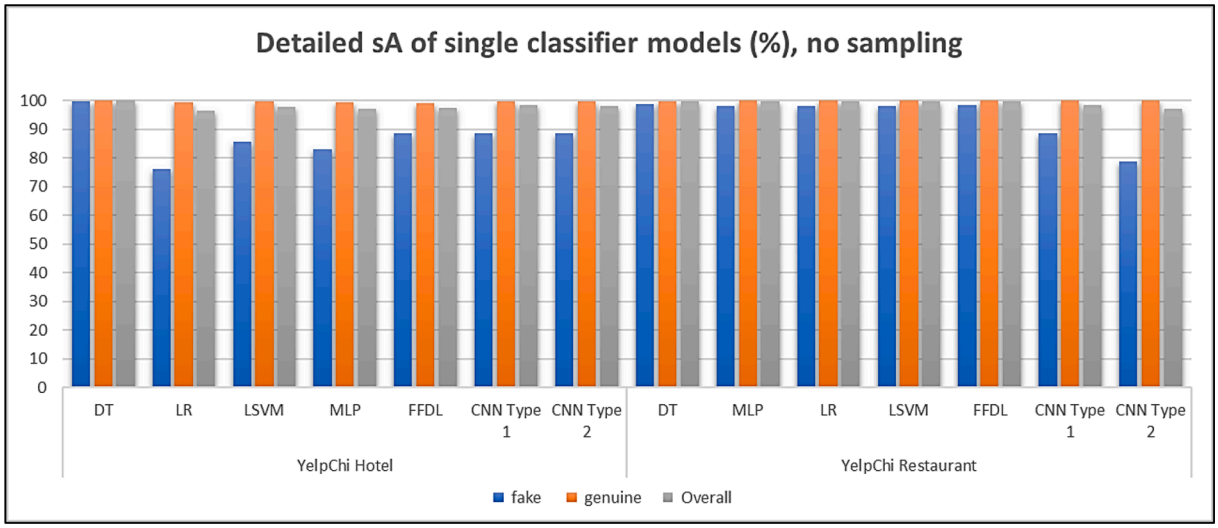

Fig. 9. Detailed sA of single classifier models for small datasets, without feature sampling.

We can see from Fig. 15 that parallel processing almost halved the processing time for the 2-fold CV. Similarly, the higher the number of *n*-fold, the higher the reduction in processing time was. For example, in the 10-fold setting, the processing time with parallel CV was more than four times less than with the iteration/normal version. However, even if run in parallel, why was the processing time of 10-fold higher than 2-fold? The difference between the processing times for different CV processes is mainly because of the difference in the amount of training data. On 2-fold, data is split equally (50% for training and 50% for testing), whereas on 4-fold, data is split into 4 sections (3 sections to train the classifier and 1 section for testing), and so on. On 10-fold CV, the amount of training data is 9/10 of the whole data, and therefore, the training time of 10-fold is the longest compared to the other settings. Another reason is the limit of CPU nodes required for the experiments. We ran 10 experiments for each fold setting, which means 20 CPU nodes were required for 2-fold, 40 CPU nodes for 4-fold, and so on. 10-fold

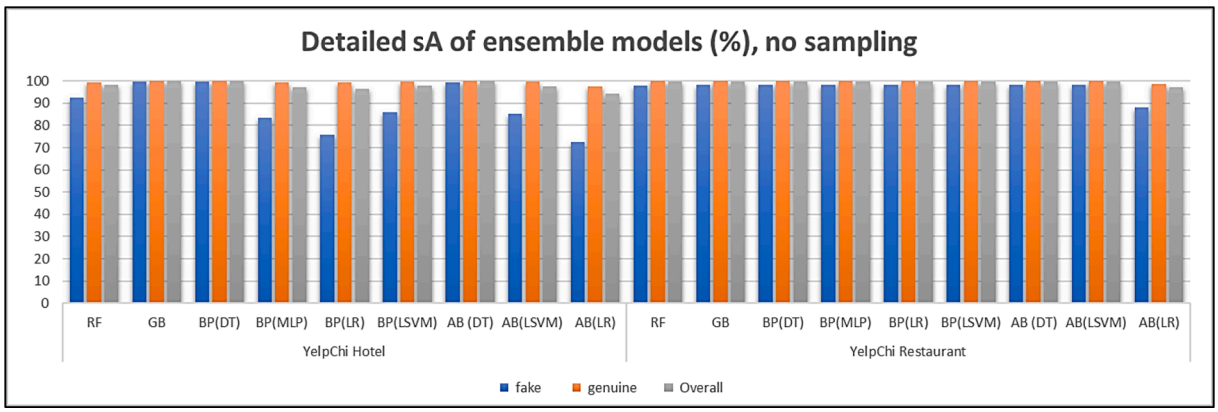

Fig. 10. Detailed sA of ensemble models for small datasets, without feature sampling.

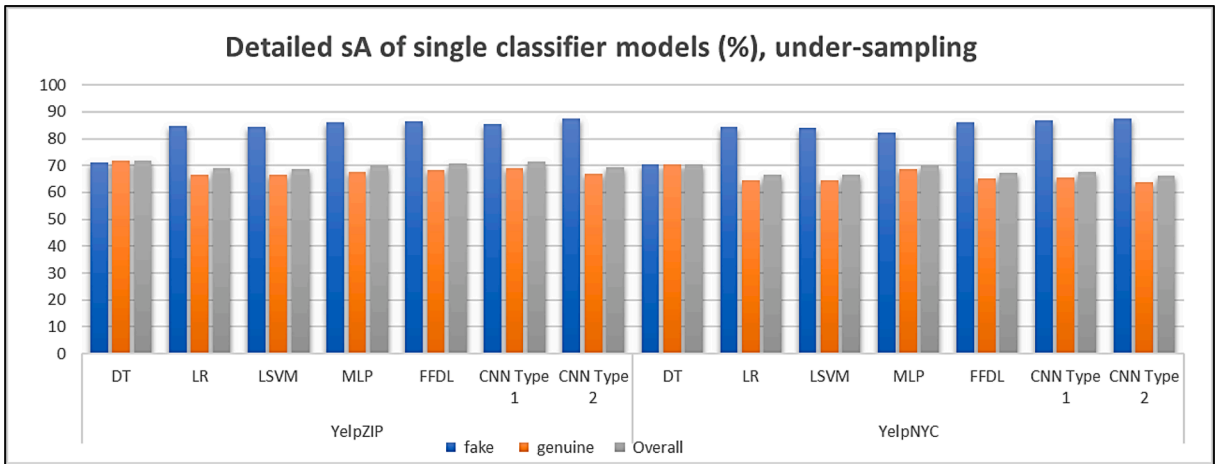

Fig. 11. Detailed sA of single classifier models for large datasets, with the under-sampling featuring approach.

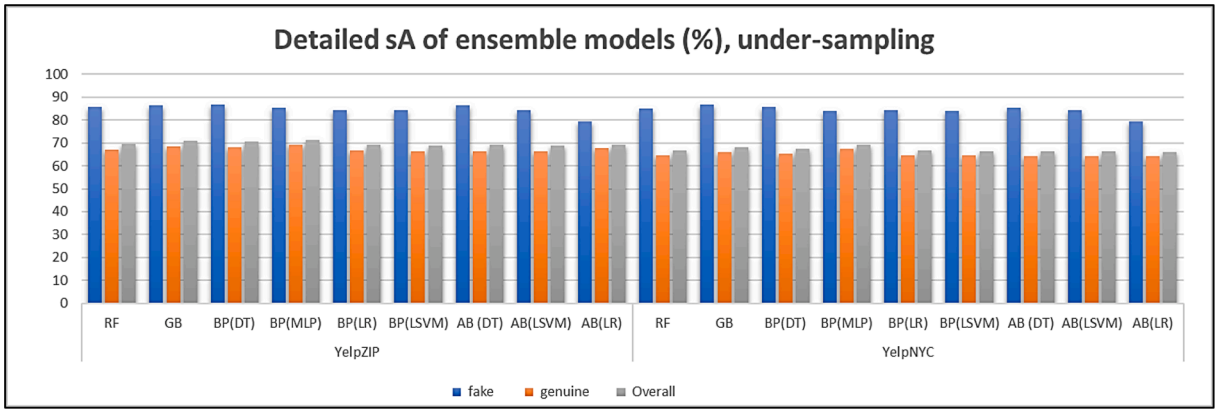

Fig. 12. Detailed sA of ensemble models for large datasets, with the under-sampling featuring approach.

ideally needs 100 nodes working together, while the limit of the cluster used in the experiments is 32 CPU nodes; hence, not all jobs could be served at the same time. In addition, these jobs had to queue together with other jobs. Therefore, the standard deviation of the processing times for 10 experiments of parallel CV is more for higher values of n.

6.5. Experiments on other datasets

To further validate the performance of our proposed approach on different datasets, we conducted experiments using 6 public datasets from the literature (see Table 1, and Table 8 for additional details). For the following experiments, the two best classifiers from the above experiments (CNN Type 1 and GB) were used on these datasets.

The datasets used by Mukherjee et al. (2013) were gathered from Yelp! and have similar attributes to those used by Rayana and Akoglu (Rayana and Akoglu, 2015). Therefore, we can apply all content-based and behaviour-based features for training (A to N in Table 3). While Mukherjee et al. did not use all the records in their datasets, we used all the records that could be extracted from the datasets. The other two public datasets, from Ott et al. (2013) and Li et al. (2014), were created

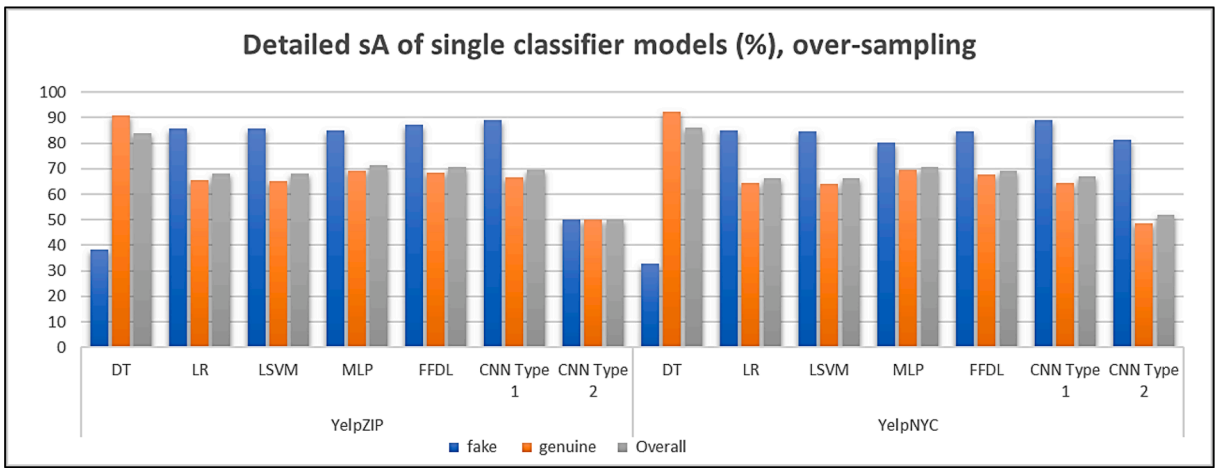

Fig. 13. Detailed sA of single classifier models for large datasets, with the over-sampling featuring approach.

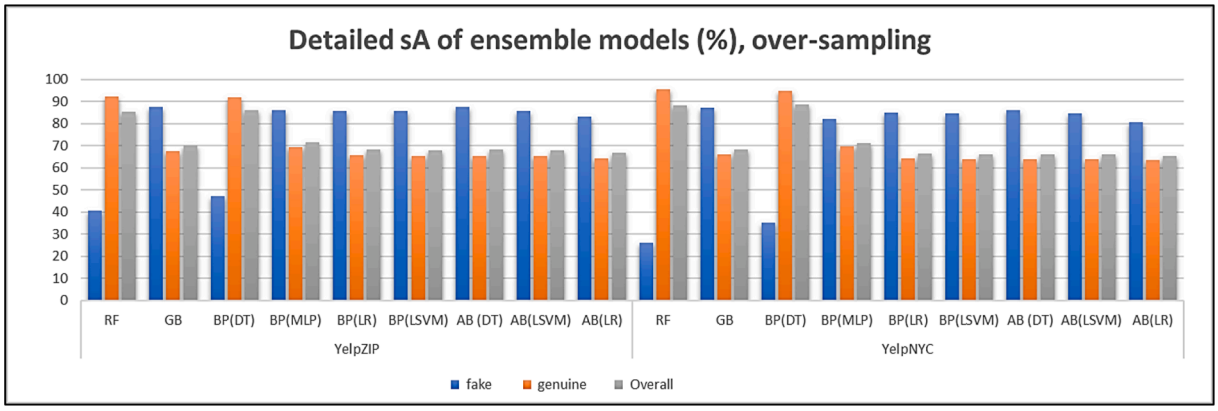

Fig. 14. Detailed sA of ensemble models for large datasets, with the over-sampling featuring approach.

by domain experts, Amazon's Mechanical Turk service (Turkers) and hotel employees to provide false/fake reviews to some online services. Li et al.'s Hotel dataset is the extension of Ott et al.'s dataset. These datasets do not have information of the reviewers, time of reviews, and the ranks/stars given. Therefore, we cannot extract our behaviour-based features, and we can only apply Section 1 – content-based featuring (A–F in Table 3) – for these datasets.

Predictably, we can see from Fig. 16 that our approach performed well on Mukherjee et al.'s datasets, because these datasets are also sourced from Yelp!. Upon further investigation, we see that our approach performed better on the Restaurant dataset than the Hotel dataset. One probable reason is that hotels have more complex services than restaurants, and therefore, the reviews for hotels are also more diverse and complex; another reason could be that chain restaurants have less positive fake reviews (Luca and Zervas, 2016). The imbalance issue is also a major factor. Mukherjee et al.'s hotel dataset is more balanced than the restaurant dataset. Better balance means more varied combination of fake review samples, which makes training and prediction more difficult. With similarly balanced datasets, such as both being imbalanced as in Rayana & Akoglu's YelpChi Hotel and Restaurant datasets (see Table 2) or both being balanced as in Li et al.'s datasets (see Table 8), the results are similar for both datasets (see Table 9; nos. 5 and 6 for Rayana & Akoglu's datasets, and nos. 8 and 9 for Li et al's datasets).

While the content-based features used in this study were not

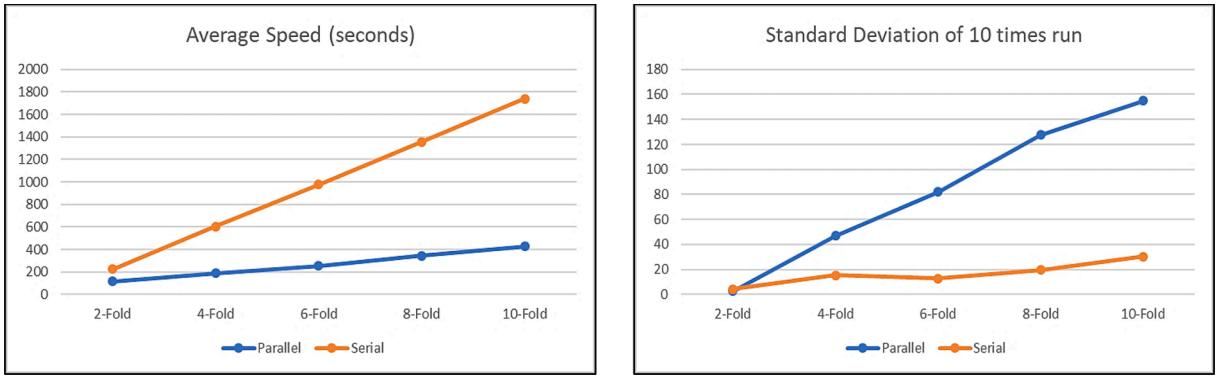

Fig. 15. Speed comparison between parallel vs. serial (iteration) processing of n-fold CV (LR, YelpZIP).

designed for standalone prediction (without behaviour-based features), they performed well with GB for all sampling settings (see Fig. 17). The accuracy of fake review class was 70% or above on all datasets. Even with over-sampling, the accuracy of fake review class in all datasets was in the range of 80–87%, except for Li et al.'s Restaurant dataset (75.1%). The performance of CNN Type 1, in general, was worse than GB on Ott et al. and Li et al.'s datasets. CNN Type 1 is probably less suited than GB for predicting fake reviews on smaller datasets, since it performed slightly worse than GB on the YelpChi Hotel and Restaurant dataset (see Figs. 9 and 10). However, it generally performed slightly better than GB on larger datasets when combined with over-sampling that makes the large datasets even larger (see Figs. 11–14, and Fig. 16).

With all experiments having used different machine/deep learning classifiers and ran on several different public fake review datasets, we can safely say that our approach can, generally, be implemented for fake review detection. The obvious limitation being that this approach has only been tested on supervised classifiers. While applying a subset of the feature groups is possible, it is suggested that – if the processed data supports it – all the feature groups (A to N) be implemented to achieve the best result (See Fig. 18).

6.6. Performance comparison with other approaches in previous studies

The performance of our approach can only be compared to other approaches with various considerations. Many factors can influence the results, such as the measurement formulas or tools, datasets used, number of records involved in experiments, and how the experiments were conducted (e.g., CV vs. traditional train-test-validation split). Therefore, the comparisons presented in Table 9 and Fig. 19 should be read with the following considerations in mind:

1. Our approach is only compared with other approaches tested on the same datasets (Tables 2 and 8). These include 10 datasets and one combination of datasets (YelpChi, which combines the YelpChi Hotel and YelpChi Restaurant datasets). We have used all samples that could be processed from each dataset. Note that some studies used only a subset of the dataset, or split the dataset into subsets and measured each subset differently. All experiments were conducted using 10-fold CV on the two best performing classifiers (CNN Type 1 and GB), with three sampling settings (no, under and over-sampling).
2. It is impossible to ensure that all the studies used the same formulas to measure their accuracy, precision, recall, F1, average precision and AUC. Therefore, our results were measured using widely used formulas for binary target prediction, which include blA, bP and bR with the fake review class as the positive class, AP and AUC. All measurements were in the "macro" average setting using scikit-learn measurement components. The F1 scores were calculated using the bF1 formula (see Table 5).
3. For comparison, we present only the best result on each dataset from each paper, including ours. The original results from the dataset creators are used as the baseline (see the underlined description and scores in Table 9, and the left-most bar group in Fig. 19). Comparison for all datasets is presented, except the baseline results for the datasets by Rayana and Akoglu (Rayana and Akoglu, 2015). The comparison with Rayana and Akoglu's results is presented separately in Fig. 19, since their results were measured using the AP and AUC, rather than traditional measures (accuracy, precision, recall and F1).

Since the measurement of results can be affected by several factors, as mentioned above, we do not claim that our proposed approach is better or worse than other approaches and let the readers draw their own conclusions.

As can be seen in the sampling process shown in Fig. 2, sampling was not conducted for the test samples. Therefore, during the testing phase, the genuine class samples overwhelm the fake class samples at a ratio of around 9:1 (see Table 2, YelpNYC and YelpZIP). Additionally, we can see in Fig. 13 that the sA of genuine class is lower than the sA of the fake class. Both the above facts mean that the CNN Type 1 model generated fp more than tp for the fake class, which is why binary precision scores for YelpNYC and YelpZIP in Table 9 are poor. However, when another formula such as wP (the weighted average precision of fake and genuine classes, see Table 5) is used to calculate the precision, the precision of CNN Type 1 (A->N) over-sampling is much better (wP of YelpNYC = 90.34%; wP of YelpZIP = 88.47%).

As discussed in the "Experiments on other datasets" section, we used only the content-based features (A to F features) for the experiments on the datasets by Ott et al. (2013) and Li et al. (2014). However, in our approach, the content-based features have not been designed to be used by themselves and without other features listed in Table 3; hence, our approach performs worse than the other approaches (as seen in Table 9 rows 7–10).

7. Conclusion and future work

Fake or fraudulent reviews on e-commerce platforms is an acute problem, which has prompted companies and researchers to make concerted efforts towards finding solutions. In this paper, we have proposed 133 different unique features from the combination of content-based and behaviour-based feature extraction approaches to be used with machine learning classifiers for detection of fake reviews. Together, these approaches can provide good results for all the datasets tested. However, fine-grained analysis reveals that the accuracies of fake and genuine classes are heavily imbalanced for the two big datasets (YelpNYC and YelpZIP) – the fake review class' accuracies are between 0.08% and 17.58%, respectively, compared to 99.99% and 97.99% for the genuine review class. We suspect the highly imbalanced data samples cause imbalanced results on big datasets.

We overcame the problem of imbalanced data using random sampling methods. For almost all classifiers, both sampling methods greatly increased the accuracy of the fake review class. However, with over-

Table 8
The statistics of several public datasets.

Dataset Author	Domain	Total record	Fake		Genuine		Content-based	Behaviour-based
			Total	%	Total	%		
Mukherjee et al. (Mukherjee et al., 2013)	Hotel (Yelp)	70,405	28,928	41.09	41,477	58.91	✓	✓
	Restaurant (Yelp)	70,500	9717	13.78	60,783	86.22	✓	✓
Ott et al. (Ott et al., 2013)	Hotel (Various)	1600	800	50.00	800	50.00	✓	
Li et al. (Li et al., 2014)	Doctor (Various)	558	357	63.98	201	36.02	✓	
	Hotel (Various)	1880	1080	57.45	800	42.55	✓	
	Restaurant (Various)	402	202	50.25	200	49.75	✓	

Table 9
Performance comparison with other studies*.

No	Domain	Description	Acc (%)	Pre (%)	Rec (%)	F1 (%)
1	Hotel Yelp (Mukherjee et al., 2013)	Mukherjee et al. (Mukherjee et al., 2013), word unigrams	65.6	62.9	76.6	68.9
		Hazim et al. (Hazim et al., 2018), with proposed features	87.43	62.96	43.97	51.78
		(Ours) *CNN Type 1 (A->N), over-sampling*	*77.23*	***68.34***	***80.53***	***73.94***
2	Restaurant Yelp (Mukherjee et al., 2013)	Mukherjee et al. (Mukherjee et al., 2013); word bigrams	67.8	64.5	79.3	71.1
		Zhang et al. (Zhang et al., 2016), RF, verbal + non-verbal feat.	83.99	86.01	89.89	87.87
		(Ours) *CNN Type 1 (A->N), under-sampling*	***97.12***	*80.22*	***98.11***	***88.27***
3	YelpNYC (Rayana and Akoglu, 2015)	Rastogi et al. (Rastogi et al., 2020), MLP on prod.-centric subset	–	–	81.86	79.74
		(Ours) *CNN Type 1 (A->N), over-sampling*	*76.81*	*22.32*	***89.04***	*35.69*
4	YelpZIP (Rayana and Akoglu, 2015)	Rastogi et al. (Rastogi et al., 2020), NB on prod.-centric subset	–	–	90.04	70.2
		(Ours) *CNN Type 1 (A->N), over-sampling*	*77.78*	*28.86*	*88.94*	*43.58*
5	YelpChi Hotel (Rayana and Akoglu, 2015)	Tang et al. (Tang et al., 2020), SVM + bfGAN	83	81.2	85.7	83.4
		Wang et al. (Wang et al., 2016), RE + PE + Bigram, 50:50 samples	86.5	84.2	89.9	87
		Wang et al. (Wang et al., 2017), RE + RRE + PRE	65.3	63.6	71.2	67.2
		You et al. (You et al., 2018), CNN + AEDA	80	83.9	74.2	78.7
		(Ours) *GB (A->N), over-sampling*	***99.93***	***99.9***	***99.8***	***99.85***
6	YelpChi Restaurant (Rayana and Akoglu, 2015)	Tang et al. (Tang et al., 2020), SVM + bfGAN	75.7	76.7	73.4	75.1
		Wang et al. (Wang et al., 2016), RE + PE + Bigram, 50:50 samples	89.9	86.8	91.8	89.2
		Wang et al. (Wang et al., 2017), , RE + RRE + PRE	62	59	78.8	67.5
		You et al. (You et al., 2018), CNN + AEDA	75.6	82.4	65.1	72.8
		(Ours) *GB (A->N), under-sampling*	***99.12***	***98.83***	***98.43***	***98.63***
7	Hotel (various) (Ott et al., 2013)	Ott et al. (Ott et al., 2013), SVM on positive sentiment subset	88.4	89.1	87.5	88.3
		Fusilier et al. (Hernández Fusilier et al., 2015), PU-L modified	–	85.2	72.8	78
		Rout et al. (Rout et al., 2016), DT	92.11	–	–	–
		Etaiwi and Naymat (Etaiwi and Naymat, 2017), SVM, all pre-proc. steps	85	51	86	–
		Zhang et al. (Zhang et al., 2018), DRI-RCNN, 0.9 T.P.	–	–	–	86.59
		(Ours) *GB (A->F), over-sampling*	*70.62*	*67.58*	*81.29*	*73.8*
8	Hotel (various) (Li et al., 2014)	Li et al. (Li et al., 2014), OvR SAGE-Unigram	81.8	81.2	84	–
		Ren and Ji (Ren and Ji, 2017), Integrated, Employee/Turker	92.6	–	–	90.1
		Li et al. (Li et al., 2017), SWNN	–	84.1	87	85
		(Ours) *GB (A->F), under-sampling*	*71.29*	*73.61*	*82.79*	*77.93*
9	Restaurant (various) (Li et al., 2014)	Li et al. (Li et al., 2014), OvR SAGE-Unigram	81.7	84.2	81.6	–
		Ren and Ji (Ren and Ji, 2017), Integrated, Customer/Turker	86.9	–	–	86.8
		Li et al. (Li et al., 2017), SWNN		83.3	88.2	81
		(Ours) *GB (A->F), over-sampling*	*72.44*	*71.23*	*75.16*	*73.14*
10	Doctor (various) (Li et al., 2014)	Li et al. (Li et al., 2014), OvR SAGE-Unigram	74.5	77.2	70.1	–
		Ren and Ji (Ren and Ji, 2017), Integrated, Customer/Turker	76	–	–	74.1
		Li et al. (Li et al., 2017), SWNN	–	83.7	87.6	82.9
		(Ours) *GB (A->F), over-sampling*	*73.35*	*79.44*	*86.94*	***83.02***

* for this table, our calculations were done using balanced-accuracy (blA), binary-precision (bP), binary-recall (bR) and binary-Fmeasure (bF1) formulas, as seen in Table 5; we provide our results in bold whenever the score is the highest.

sampling, we found that the performance of DT-type classifiers (such as the DT itself, RF and BP) is not as good as other classifiers. However, the combination of boosting methods in AB and DT provides comparable performance with other classifiers. Best results on the fake review class' accuracy for both big datasets were achieved by the CNN Type 1 and GB ensemble, which also performed well with the smaller datasets.

We also proposed a parallel-CV method, which has the potential to highly reduce the time required for the n-fold CV process. Our experimental results revealed that the proposed method could halve the time required for 2-fold CV and provide even more reduction for higher values of n, such that the time required for 10-fold CV is less than a fourth of the normal processing time.

From the results of testing our approach on other datasets, we observed that our proposed hybrid approach performed well on those datasets too. The results of experiments on Mukherjee et al.'s datasets also informed us that our approach performs better on imbalanced datasets than balanced datasets. In addition, we found that, in the case of GB ensemble with over-sampling, our content-based features (Groups A to F) work well on smaller datasets and provide fake review class accuracies between 75 and 87%.

Summing up, this work contributes theoretically to the literature of both natural language processing and machine/deep learning, and practically to online commerce security problems. Our findings provide insights on processing text materials, approaches to extracting text features, and detection of fraudulent online customer reviews. Reliable and effective detection of fake reviews will increase the trustworthiness of online commerce, and therefore, detection and elimination of fake reviews is a high priority and urgent goal of online commerce portal providers.

Despite the extensive experimental studies presented in this paper,

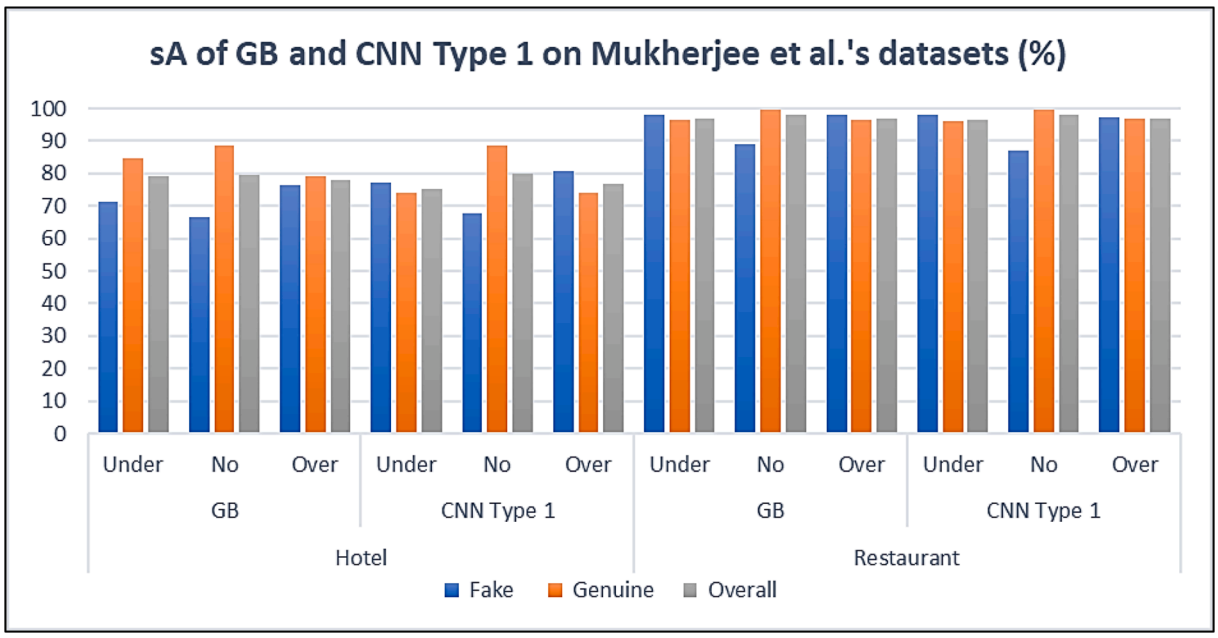

Fig. 16. sA of applying all features (A–N) on GB and CNN Type 1 classifiers on Mukherjee et al.'s datasets.

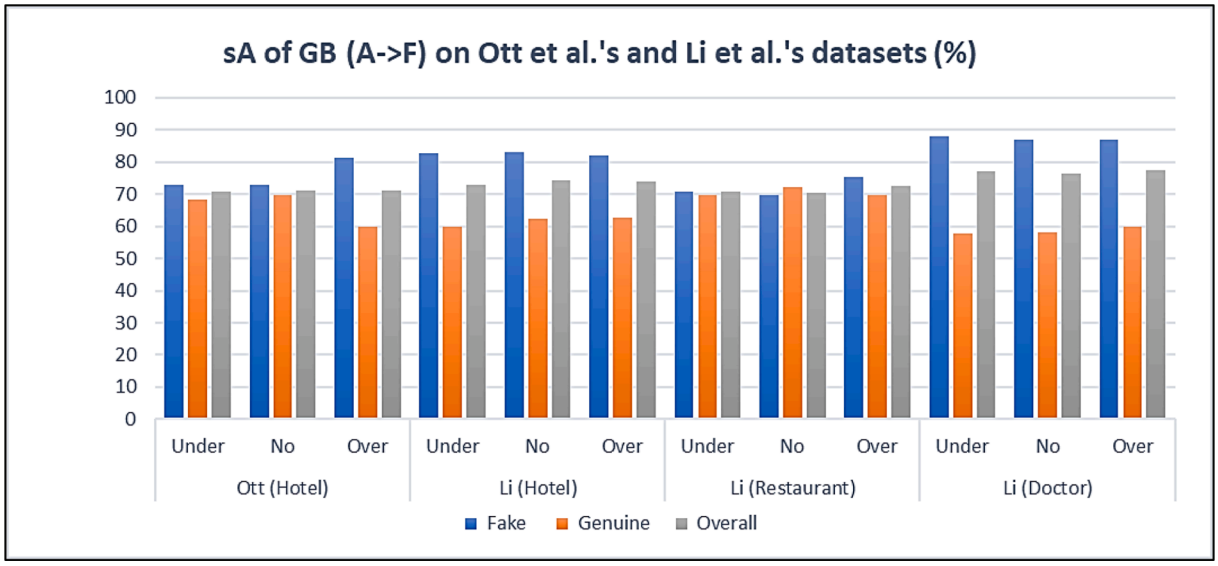

Fig. 17. sA of GB using content-based features (A–F) on Ott et al. and Li et al.'s datasets.

there are still opportunities/possibilities for further improvement. In our future work, we plan to investigate a novel idea about the combination of multiple classifiers and the combination of multiple approaches of feature extraction for improving accuracy on larger datasets.

CRediT authorship contribution statement

Gregorius Satia Budhi: Conceptualization, Formal analysis, Formal analysis, Investigation, Methodology, Validation, Writing - review & editing. **Raymond Chiong:** Conceptualization, Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing. **Zuli Wang:** Investigation, Methodology. **Sandeep Dhakal:**

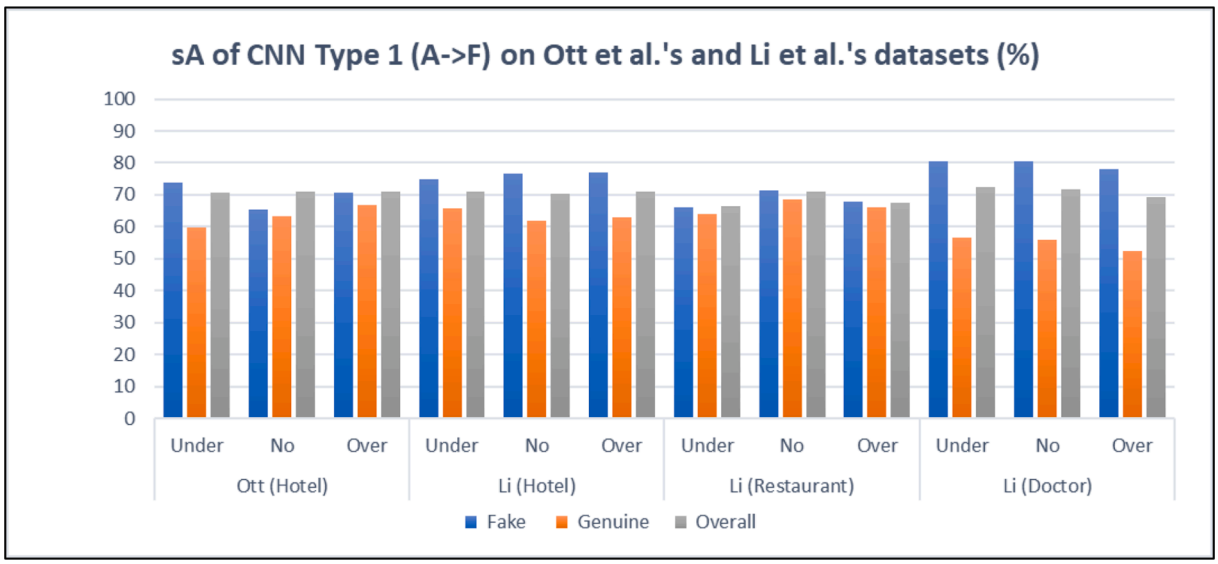

Fig. 18. sA of CNN Type 1 using content-based features (A–F) on Ott et al. and Li et al.'s datasets.

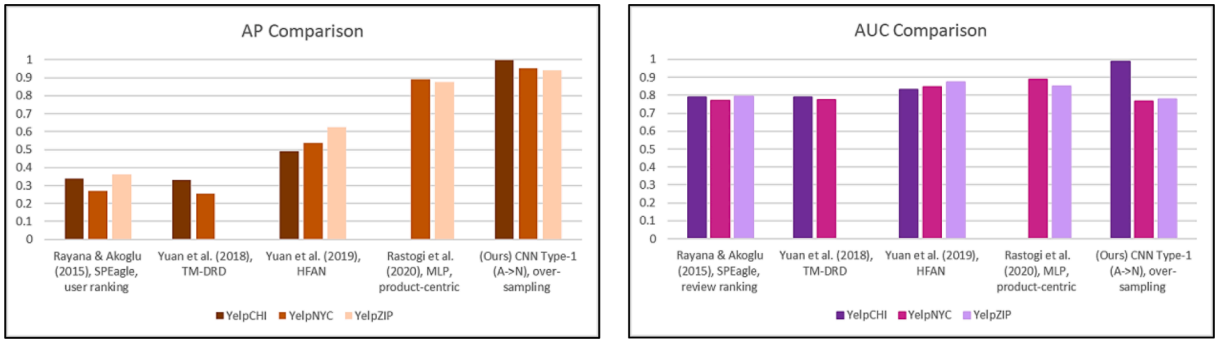

Fig. 19. AP and AUC score comparison for YelpCHI, YelpNYC and YelpZIP datasets.

Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

Utz, S., Kerkhof, P., van den Bos, J., 2012. Consumers rule: how consumer reviews influence perceived trustworthiness of online stores. Electron. Commer. Res. Appl. 11 (1), 49–58. https://doi.org/10.1016/j.elerap.2011.07.010.

Bagheri, A., Saraee, M., de Jong, F., 2013. Care more about customers: Unsupervised domain-independent aspect detection for sentiment analysis of customer reviews. Knowl.-Based Syst. 52, 201–213. https://doi.org/10.1016/j.knosys.2013.08.011.

Bajaj, S., Garg, N., Singh, S.K., 2017. A novel user-based spam review detection. Procedia Comput. Sci. 122, 1009–1015.

Budhi GS, Chiong R, Pranata I, Hu Z Predicting rating polarity through automatic classification of review texts. In: Proceedings of the 2017 IEEE Conference on Big Data and Analytics (ICBDA), Kuching, Malaysia, November 16-17, 2017. pp 19-24. doi:10.1109/ICBDAA.2017.8284101.

Feng VW, Hirst G Detec.ting deceptive opinions with profile compatibility. In: Proceedings of International Joint Conference on Natural Language Processing, Nagoya, Japan, October 14-18, 2013. pp. 338–346.

Jindal N, Liu B Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining Palo Alto, California, USA, February 11-12, 2008. pp 219–230.

Song, W., Li, W., Geng, S., 2020. Effect of online product reviews on third parties' selling on retail platforms. Electron. Commer. Res. Appl. 39, 100900. https://doi.org/10.1016/j.elerap.2019.100900.

Felbermayr, A., Nanopoulos, A., 2016. The role of emotions for the perceived usefulness in online customer reviews. J. Interact. Market. 36, 60–76. https://doi.org/10.1016/j.intmar.2016.05.004.

Mukherjee A, Kumar A, Liu B, Wang J, Hsu M, Castellanos M, Ghosh R Spotting Opinion Spammers using Behavioral Footprints. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago Illinois USA, August 11-14, 2013. pp. 632–640.

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.

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.

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.

Cardoso, E.F., Silva, R.M., Almeida, T.A., 2018. Towards automatic filtering of fake reviews. Neurocomputing 309, 106–116. https://doi.org/10.1016/j.neucom.2018.04.074.

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 October 02 2019.

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.

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.

Luca, M., Zervas, G., 2016. Fake it till you make it: Reputation, competition, and Yelp review fraud. Manage. Sci. 62 (12), 3412–3427. https://doi.org/10.1287/mnsc.2015.2304.
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.
Ott, M., Choi, Y., Cardie, C., Hancock, J.T., 2011. Finding deceptive opinion spam by any stretch of the imagination 19–24, 309–319.
Salehan, M., Kim, D.J., 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.
Fang, J., Hu, L., Hossin, M.A., Yang, J., Shao, Y., 2019. Polluted online reviews: the effect of air pollution on reviewer behavior. Int. J. Electron. Comm. 23 (4), 557–594. https://doi.org/10.1080/10864415.2019.1655206.
Barbado, R., Araque, O., Iglesias, C.A., 2019. A framework for fake review detection in online consumer electronics retailers. Inf. Process. Manage. 56 (4), 1234–1244. https://doi.org/10.1016/j.ipm.2019.03.002.
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.
Hernández Fusilier, D., Montes-y-Gómez, M., Rosso, P., Guzmán Cabrera, R., 2015. Detecting positive and negative deceptive opinions using PU-learning. Inf. Process. Manage. 51 (4), 433–443. https://doi.org/10.1016/j.ipm.2014.11.001.
Etaiwi, W., Naymat, G., 2017. The impact of applying different preprocessing steps on review spam detection. Procedia Comput. Sci. 113, 273–279.
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.
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 Transactions on Internet and Information Systems 12 (10). doi: 10.3837/tiis.2018.10.026.
Rayana, S., Akoglu, L., 2015. Collective opinion apam setection: Bridging review networks and metadata. In: In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 985–994. https://doi.org/10.1145/2783258.2783370.
You Z, Qian T, Liu B An Attribute Enhanced Domain Adaptive Model for Cold-Start Spam Review Detection. In: Proceedings ofthe 27th International Conference on Computational Linguistics, Santa Fe, New Mexico, USA, August 20-26, 2018. pp 1884-1895.
Yuan C, Zhou W, Ma Q, Lv S, Han J, Hu S Learning review representations from user and product level information for spam detection. In: Proceedings of 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8-11 November, 2019. pp 1–6.
Yuan, L., Li, D., Wei, S., Wang, M., 2018. Research of deceptive review detection based on target product identification and metapath feature weight calculation. Complexity 2018, 1–12. https://doi.org/10.1155/2018/5321280.
Rastogi, A., Mehrotra, M., Ali, S.S., 2020. Effective opinion spam detection: a study on review metadata versus content. J. Data Inform. Sci. 5 (2), 76–110. https://doi.org/10.2478/jdis-2020-0013.
Tang, X., Qian, T., You, Z., 2020. Generating behavior features for cold-start spam review detection with adversarial learning. Inf. Sci. 526, 274–288. https://doi.org/10.1016/j.ins.2020.03.063.
Sun, C., Du, Q., Tian, G., 2016. Exploiting product related review features for fake review detection. Mathemat. Probl. Eng. 2016, 1–7. https://doi.org/10.1155/2016/4935792.
Zhang, W., Du, Y., Yoshida, T., Wang, Q., 2018. DRI-RCNN: an approach to deceptive review identification using recurrent convolutional neural network. Inf. Process. Manage. 54 (4), 576–592. https://doi.org/10.1016/j.ipm.2018.03.007.
Ott M, Cardie C, Hancock JT Negative deceptive opinion spam. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, Georgia, US, June 9-14, 2013. pp. 497–501.
Li J, Ott M, Cardie C, Hovy E Towards a general rule for identifying deceptive opinion spam. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, USA, June 23-25, 2014. pp 1566-1576.
Budhi, G.S., Chiong, R., Pranata, I., Hu, Z., 2021. Using machine learning to predict the sentiment of online reviews: a new framework for comparative analysis. Arch. Comput. Methods Eng. https://doi.org/10.1007/s11831-020-09464-8.
Rout, J.K., Singh, S., Jena, S.K., Bakshi, S., 2016. Deceptive review detection using labeled and unlabeled data. Multimedia Tools Appl. 76 (3), 3187–3211. https://doi.org/10.1007/s11042-016-3819-y.
Zhang, D., Zhou, L., Kehoe, J.L., Kilic, I.Y., 2016. What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fake online reviews. J. Managem. Inform. Syst. 33 (2), 456–481. https://doi.org/10.1080/07421222.2016.1205907.
Wahyuni ED, Djunaidy A Fake review detection from a product review using modified method of iterative computation framework. In: Proceedings of MATEC Web of Conferences 58, 03003, 2016. doi:10.1051/matec.
Heydari, A., Tavakoli, M., Salim, N., 2016. Detection of fake opinions using time series. Expert Syst. Appl. 58, 83–92. https://doi.org/10.1016/j.eswa.2016.03.020.
Wang X, He KLS, Zhao J Learning to Represent Review with Tensor Decomposition for Spam Detection. In: Proceedings ofthe 2016 Conference on Empirical Methods in Natural Language Processing, Texas, US, November 1-5, 2016. pp. 866–875.
Hazim, M., Anuar, N.B., Ab Razak, M.F., Abdullah, N.A., Emmert-Streib, F., 2018. Detecting opinion spams through supervised boosting approach. PLoS ONE 13 (6), e0198884. https://doi.org/10.1371/journal.pone.019884.
Rathore, S., Loia, V., Park, J.H., 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.
Li, H., Chen, Z., Mukherjee, A., Liu, B., Shao, J., 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns 26–29, 634–637.
Kumar, N., Venugopal, D., Qiu, L., Kumar, S., 2018. Detecting review manipulation on online platforms with hierarchical supervised learning. J. Manage. Inform. Syst. 35 (1), 350–380. https://doi.org/10.1080/07421222.2018.1440758.
Dong, M., Yao, L., Wang, X., Benatallah, B., Huang, C., Ning, X., 2020. Opinion fraud detection via neural autoencoder decision forest. Pattern Recogn. Lett. 132, 21–29. https://doi.org/10.1016/j.patrec.2018.07.013.
Martens, D., Maalej, W., 2019. Towards understanding and detecting fake reviews in app stores. Empir. Softw. Eng. 24 (6), 3316–3355. https://doi.org/10.1007/s10664-019-09706-9.
Wang X, Liu K, Zhao J 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 (Volume 1: Long Papers), Vancouver, Canada, July 30-August 4, 2017. pp 366-376. doi:10.18653/v1/P17-1034.
Mukherjee A, Venkataraman V, Liu B, Glance N What Yelp Fake Review Filter Might Be Doing? In: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media, Boston, USA, July 8-10, 2013. pp 409-418.
Yelp (2019) Yelp dataset challenge: Round 13. https://www.yelp.com/dataset/challenge. Accessed December 27 2019.
NLTK (2019) Nltk Package. http://www.nltk.org/api/nltk.html. Accessed January 25 2019.
Bansal S, Aggarwal C (2019) textstat 0.5.6. https://pypi.org/project/textstat/#description. Accessed October 2 2019.
Buchholz, S., 2002. Memory-Based Grammatical Relation Finding. Tilburg, Eigen beheer.
Shuteyev P (2018) 550+ spam trigger words to avoid in 2019. https://snov.io/blog/550-spam-trigger-words-to-avoid-in-2019/. 2019.
Perelsztejn F (2017) 455 spam trigger words to avoid in 2019. https://blog.prospect.io/455-email-spam-trigger-words-avoid-2018/. 2019.
Pels H (2019) 200+ spam trigger keywords to avoid in your emails. https://www.emarsys.com/resources/blog/email-spam-keywords-to-avoid/. 2019.
Baccianella S, Esuli A, Sebastian F SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of International Conference on Language Resources and Evaluation (LREC), Valletta, Malta, May 17-23, 2010. pp 2200-2204.
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. Indus. Manage. Data Syst. 119 (3), 676–696. https://doi.org/10.1108/IMDS-02-2018-0072.
Budhi, G.S., Chiong, R., Wang, Z., 2021. Resampling imbalanced data to detect fake reviews using machine learning classifiers and textual-based features. Multimedia Tools Appl. https://doi.org/10.1007/s11042-020-10299-5.
Lo, S.L., Cambria, E., Chiong, R., Cornforth, D., 2017. Multilingual sentiment analysis: From formal to informal and scarce resource languages. Artif. Intell. Rev. 48 (4), 499–527. https://doi.org/10.1007/s10462-016-9508-4.
Lo, S.L., Chiong, R., Cornforth, D., 2015. Using support vector machine ensembles for target audience classification on Twitter. PLoS ONE 10 (4), e0122855.
Lo, S.L., Chiong, R., Cornforth, D., 2016. Ranking of high-value social audiences on Twitter. Decis. Support Syst. 85, 34–48. https://doi.org/10.1016/j.dss.2016.02.010.
Hu Z, Chiong R, Pranata I, Susilo W, Bao Y Identifying malicious web domains using machine learning techniques with online credibility and performance data. In: Proceedings of Congress on Evolutionary Computation (CEC), Vancouver, Canada, July 24-29, 2016. pp. 5186–5194.
Menard, S., 2010. Logistic Regression: From Introductory to Advanced Concepts and Applications. SAGE, Los Angeles.
Campbell, Colin, Ying, Yiming, 2011. Learning with Support Vector Machines. Morgan & Claypool. 5 (1), 1–95.
Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. 2 (3), 1–27. https://doi.org/10.1145/1961189.1961199.
Glorot X, Bengio Y Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, May 13-15, 2010. pp 249-256.

Kingma DP, Ba J Adam: A method for stochastic optimization. In: Proceedings of International Conference on Learning Representations, San Diego, US, May 7-9, 2015. pp 1-15.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol 1. MIT Press, pp. 318–362.
Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1 (1), 81–106. https://doi.org/10.1007/bf00116251.
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32. https://doi.org/10.1023/a:1010933404324.
Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Statist. 29 (5), 1189–1232.
Breiman, L., 1996. Bagging predictors. Machine Learning 24 (2), 123–140. https://doi.org/10.1007/bf00058655.
Zhu, J., Zou, H., Rosset, S., Hastie, T., 2009. Multi-class adaboost. statistics and its. Interface 2, 349–360.
Yu, Yuhai, Lin, Hongfei, Meng, Jiana, Zhao, Zhehuan, 2016. Visual and textual sentiment analysis of a microblog using deep convolutional neural networks. Algorithms 9 (2), 41. https://doi.org/10.3390/a9020041.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90. https://doi.org/10.1145/3065386.
Lee, S., Ha, J., Zokhirova, M., Moon, H., Lee, J., 2017. Background information of deep learning for structural engineering. Arch. Comput. Methods Eng. 25 (1), 121–129. https://doi.org/10.1007/s11831-017-9237-0.
Scikit-learn (2019) API Reference. http://scikit-learn.org/stable/modules/classes.html. Accessed March 19 2019.
Keras (2019) Keras: The Python Deep Learning library. https://keras.io/. Accessed March 8 2019.