International Journal of Computational Intelligence Systems
Vol. **12(1)**; 2019, *pp.* 387–394
DOI: https://doi.org/10.2991/ijcis.2019.125905655; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Enactment of Ensemble Learning for Review Spam Detection on Selected Features

Faisal Khurshid*, Yan Zhu, Zhuang Xu, Mushtaq Ahmad, Muqeet Ahmad

School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China

ARTICLE INFO

Article History
Received 15 Apr 2018
Revised 06 Jan 2019
Accepted 11 Jan 2019

Keywords
Review spam
Ensemble learning module
Positive polarity
Negative polarity

ABSTRACT

In the ongoing era of flourishing e-commerce, people prefer online purchasing products and services to save time. These online purchase decisions are mostly influenced by the reviews/opinions of others who already have experienced them. Malicious users use this experience sharing to promote or degrade products/services for their iniquitous monetary benefits, known as review spam. This study aims to evaluate the performance of ensemble learning on review spam detection with selected features extracted from real and semi-real-life datasets. We study various performance metrics including Precision, Recall, F-Measure, and Receiver Operating Characteristic (RoC). Our proposed ensemble learning module (ELM) with ChiSquared feature selection technique outperformed all others with 0.851 Precision.

1. INTRODUCTION

In the pre-Internet era for decision-making people used to discuss, exchange their opinions/views with friends, relatives, or consulted consumer reports. However, in this post-Internet era, people use to consult review websites, online forums, and e-commerce sites, where consumers share their opinions, views, and experiences commonly known as reviews. These reviews influence potential buyer's buying decision about a particular product, service, or trend. The aim of these reviews is to help consumers in making the buying decision easier. These reviews are used as tool; for improvement in products and services benchmarking, market intelligence strategies, and also for policy making by businesses and organizations [1].

Targeted products or services may be promoted or degraded by mischievous users using these reviews. Therefore, the integrity of reviews is questionable. According to Luca and Zervas [2], the revenue increases to 5–9% by a single star rising in rating. Companies or individuals hire spammers to boost their products/services or degrade the rival's products/services because of enormous monetary benefits; such mischievous activity is known as review spamming [3, 4].

Web spam, Email spam, Blog Spam, and Review spam are among several types of spam. Other types of spams are relatively easier to detect than review spam because fake reviews are written so trickily that they guise genuine. Subsequently, reviews have few contents due to their domain limitation to products/services and people's opinions; that results in fewer features which becomes hard to identify. Therefore, it is very tough to capture the sentiment analysis, domain knowledge, structural and linguistic features. Fusilier *et al.* [5] stated that spam reviews written in negative polarity are much harder to detect than the spam reviews written in positive polarity. Two reviews are shown as follows:

- *I hanged about here for a week during a conference and the service was excellent. In addition to the excellent service the location was great. With quick access to Michigan Ave. The bars and Shula's restaurant were excellent. You must have the rib-eye steak...very well done.*
- *Hotel Allegro Chicago is a beautiful place. The service there was amazing, everybody was very friendly. When I saw my room, once again my jaw almost dropped. The room was fit for a king; the designs on the wall were so elegant. The beds were extraordinarily comfy. I think for my next business trip I am going to bring my wife and kids with me. I give this Kimpton Hotel 5/5 star!*

These reviews have been taken from publically available dataset of Ott *et al.* [6]. Review one is the genuine review, whereas review two is a spam. These two reviews depict that it is very hard for readers to distinguish between genuine and spam reviews.

The spam review can be categorized into three types [4]: deceptive (fake) reviews, reviews about brands only, and nonreviews. Reviews written without any experience are deceptive (fake) reviews; their only objective is to promote or degrade a product/service. The second one instead of a specific product/service for which the review intends to be given, targets a company or a brand. The third type contains inapt discussions, advertisements, and hyperlinks to some

*Corresponding author. Email: faisalnit@gmail.com

other product/service pages, and so on. It is harder to detect the first type than the other two types, where the other two can easily be detected by applying different techniques or simply by reading.

Review spam detection is actually the process of cataloging reviews into genuine and spam reviews. The scarcity of labeled dataset in enormous volume makes it challenging to apply supervised machine learning techniques to achieve accurate results; even though many studies have been carried out that we discuss in Section 2.

A small semi-real polarity-based dataset having only 1600 reviews (800 each in positive and negative polarity) is developed by Ott *et al.* [6, 7]; out of which in positive polarity the 400 genuine reviews were crawled from tripadvisor.com and 400 spam reviews were tailored by hiring Mechanical Turkers [6]; whereas for negative polarity they crawled 400 genuine reviews from different review websites like tripadvisor.com, priceline.com, hotels.com, expedia.com, and orbitz.com; and again 400 spam reviews were tailored by Mechanical Turkers [7]. Many researchers like Feng *et al.* [8], Li *et al.* [9], Fusilier *et al.* [10], and so on, used the same Ott's dataset for their respective studies.

This paper contributes to the literature in several ways; this study is the extension of our previous work Khurshid *et al.* [11]. Firstly, we have also used a real-life data set instead of only using the semi-real dataset as just using the semi-real dataset is not trustworthy Mukherjee *et al.* [12]. To the best of our knowledge, very few researchers like Mukherjee *et al.* [12], all others have used semi-real dataset, where detection of spam is comparatively easier. Secondly, we have applied ensemble learning module (ELM) instead of single base classifiers as ELM can detect review spam efficiently compare to single base classifiers. The proposed ELM incorporates four classifiers: Discriminative Multinomial Naive Bayes (DMNBtext), J48, LibSVM, and linear regression (LR). We also performed the comparative performance of our proposed system with the other state of the art base classifiers like multilayer perceptron (MLP), Naïve Bayes (NB), and Adaboost. To reduce the biases of the classifiers we use 10-fold cross-validation with proper data split (training and testing). Thirdly, we have combined features selection technique with ELM which enhances the performance of ELM by eliminating redundant and low-level features from the features vector space. Different feature sets are excerpted from the real-life dataset of Yelp.com crawled by Mukherjee *et al.* [12] and semi-real dataset of Ott *et al.* [6, 7]. Features selection technique and ensemble learning is rarely used in spam detection as highlighted by Crawford *et al.* [13].

The rest of the paper is organized as follows: Section 2 covers literature review in brief. Methodology, a detailed description of the proposed ELM, features selection techniques, experimental setup, data preprocessing, and features extraction has been discussed in Section 3. Section 4 illustrates results and their discussion. The conclusion and recommendations to future researchers are addressed in Section 5.

2. RELATED WORK

In this section we briefly discuss some previous studies held out for review spam detection. The problem of review spam was initially addressed by Jindal and Liu [4] to the best of our knowledge. The primary focus of review spamming is to boost or degrade a service or product for iniquitous monetary benefits. Hence, similar reviews may exist repeatedly for a specific service or product. In their study [4], they declared full and 90% duplicates as spams reviews. They categorized duplicates into three types: duplicates from same User ID on different products, duplicates from different User IDs on the same product, and duplicates from different User IDs on several products. They used logistic regression model and obtained 78% area under the curve (AUC) value.

Ott *et al.* [6] in their initial study developed a small dataset of 800 reviews out of which 400 were genuine and the remaining 400 spam were tailored by Amazon Mechanical Turkers (AMT) in positive polarity; they achieved an accuracy of 89% by using only linguistic features to determine spam reviews. In their another study [7], Ott *et al.* constructed another small dataset in negative polarity in same way having 400 genuine and 400 spam tailored by AMT. Feng *et al.* [8] used syntactic stylometry for spam detection. They achieved 91.2% accuracy by using features driven from contextfree grammar (CFG), where the parse trees consistently improve the detection rate. However, their experiments were based on semi-real dataset developed by Ott *et al.* [6].

Li *et al.* [9] proposed Topic Spam, a generative Latent Dirichlet Allocation (LDA)–based topic modeling approach; a detection technique for review spam. They achieved 95% accuracy from the same relatively smaller semi-real dataset of Ott *et al.* [6]. In their other work, Li *et al.* [14] proposed a general rule to sort out language differences between genuine and spam reviews by extending Bayesian generative model known as SAGE that covered multiple aspects. This study was based on a new gold standard dataset developed by Li *et al.*; having reviews from three different categories: domain expert deceptive review spam (Employee), crowdsourced deceptive review spam (Turker), and truthful customer reviews (Customer).

Wang and Zhu [15] used the Latent Semantic Indexing (LSI) and Sprinkled LSI techniques for dimension reduction. Several methods were integrated with these techniques using a voting scheme for classification and achieved 95% accuracy. Based on readability, type, and writing style of review, Banerjee and Chua [16] developed a linguistic framework to distinguish between original and spam reviews.

Mukherjee *et al.* [12, 17] argues that semi-real Ott's datasets are not trustworthy because the deceptive reviews are tailored by AMT which have limited vocabulary and almost same writing style. They achieved only 67.6% accuracy by applying the same settings of Ott *et al.* [6] on the real-life dataset of Yelp.com. They achieved 86% accuracy by using behavioral features from Yelp crawled dataset and inferred that Yelp filter might also be using behavioral features to combat review spam.

Deep level linguistic features including inter-sentence information and sentiment analysis were used by Chen *et al.* [18]. Ren and Ji. [19] developed a neural network model to learn document-level representation. Initially the model learns sentence representation with convolutional neural network. Then, sentence representations were combined using a gated recurrent neural network, which modeled discourse information and yielded a document vector. They directly used these document representations as features to identify deceptive opinion spam. Fusilier *et al.* [10] utilized N-grams features and achieved relatively good results, that is, 0.80 F1 by using

only 25% of training data. Xie *et al.* [20] worked on detecting singleton review spam (where a reviewer wrote only one review) by constructing a multidimensional time series through hierarchical detection criteria. Jindal *et al.* [21] identified review spammers by defining unexpected rules (One-Condition and Two-Condition) to identify unusual review patterns in product rating which can represent the suspicious behaviors of reviewers.

Li *et al.* [22, 23] used positive unlabeled learning (PU) for review spam detection. They used a dataset from Chinese review hosting website www.dianping.com. It was the first attempt of any study related to a Chinese website and its filtering system. They first developed a supervised learning algorithm multi-typed heterogeneous collective classification (MHCC) for the heterogeneous network of reviews, users, and IPs. Some spam reviews may still exist after filtering. To cope up with this issue, they extended MHCC to collective positive unlabeled learning (CPU) model which is more appropriate for PU learning. They claimed that this approach detects a large number of potential spam reviews hidden in the unlabeled dataset that Dianping failed to detect, which is a very positive aspect of PU learning.

Building on these studies, we tried to perform review spam detection on real and semi-real-life dataset using ELM combined with features selection techniques, the details are discussed in the proceeding sections of our paper.

3. METHODOLOGY

There are two main settings of our experiments. In setting one, we use full feature sets whereas in setting two we applied different feature selection techniques to evaluate their impact on the comparative performance of base classifiers and our proposed ELM. The main framework of our approach for review spam detection is shown in Figure 1.

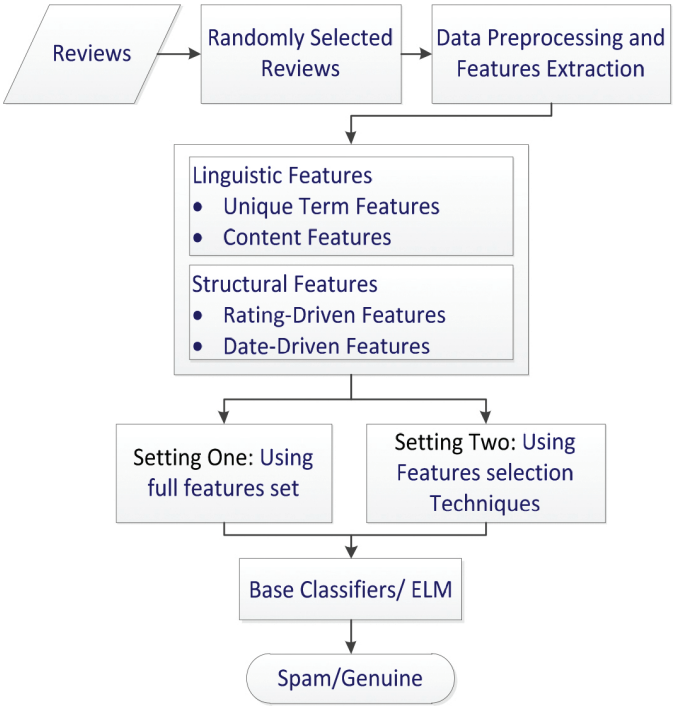

Figure 1 | System main framework.

The main objective of this study is to demonstrate ensemble learning performance for spam reviews detection and the positive effect of different feature selection techniques.

To achieve our objective, we define the following hypothesis:

Hypothesis 1: Single classifiers have inconsistent and squat performance on redundant and low-level features.

Hypothesis 2: The proposed ELM outperforms single classifiers with different feature selection techniques for review spam classification.

3.1. Ensemble Learning

Our proposed ELM is based on two tiers, where Tier 1 has three classifiers DMNB, J48, and LibSVM. LR acts as a meta-classifier in Tier 2. The output of Tier 1 classifiers is input for Tier 2 meta-classifier that corrects the improper training of Tier 1 classifiers to produce accurate results; Figure 2 depicts the proposed ELM model. To validate the enactment of our proposed system, we used state of the art base classifiers like MLP, NB, and Adaboost to compare performance.

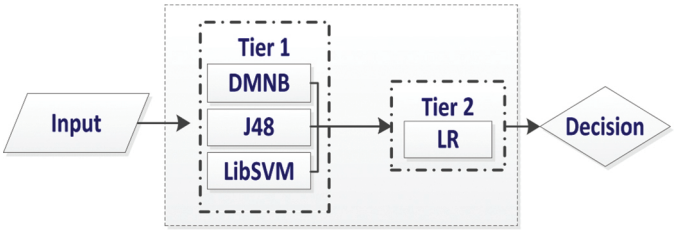

Figure 2 | Proposed ensemble learning module (ELM).

3.1.1. Multilayer perceptron

An MLP is a feedforward artificial neural network that generates a set of outputs from a set of inputs. MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses backpropagation for training the network. To improve speed, an approximate version of the logistic function is used as the default activation function for the hidden layer, but other activation functions can be specified. We used the Approximate Sigmoid function as the activation function with two hidden layers; the ridge parameter value set to 0.01 and optimized the loss function by SquaredError.

3.1.2. Naïve Bayes

NB is a simple technique for constructing models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set [24].

$$p\left(C_k|\mathbf{x}\right) = \frac{p\left(C_k\right)p\left(\mathbf{x}|C_k\right)}{p\left(\mathbf{x}\right)} \tag{1}$$

NB is a conditional probability model: given a problem instance to be classified, represented by a vector $x = (x1, \ldots, x n)$ representing some n features (independent variables), it assigns to this instance probabilities$p\left(C_k|x_1, \ldots, x_n\right)$ for each of K possible outcomes or classes C_k. We use Kernel estimator for numeric attributes rather than a normal distribution with batch size 100 for our experiments.

3.1.3. Adaboost

AdaBoost is meta-algorithm that is sensitive to noisy data and outliers, tackles nominal class problems. It often dramatically improves performance, but sometimes overfits [25]. We set random seed number to 1 and batch size to 100 with 10 iterations.

3.1.4. DMNBtext

DMNB classifier learns a multinomial NB classifier in a combined generative and discriminative fashion. DMNBtext injects a discriminative element into parameter learning by considering the current classifier's prediction for a training instance before updating frequency counts. When processing a given training instance, the counts are incremented by one minus the predicted probability for the instance's class value [26].

3.1.5. J48

The decision tree algorithm, J48, performs classification by creating a decision tree based on features of the input training data [27]. The root node of the tree is the feature with the highest information gain, that is, it has the maximum classification power. The leaf node describes the decision of the algorithm. Hence, the value of the leaf node is dependent on other independent nodes of the tree. We use the confidence factor to 0.25 and 2 as the minimum number of instances per leaf.

3.1.6. LibSVM

A library for support vector machines (SVMs). LibSVM implements the SMO algorithm for Kernelized SVM that supports classification and regression [28]. We use the kernel type linear: $u' * v$, keeping the tolerance of the termination criterion eps to 0.001.

3.1.7. Linear regression

It uses the Akaike Information Criterion (AIC) [29] for model selection, and is able to deal with weighted instances. The AIC is an estimator of the relative quality of statistical models for a given set of data. AIC estimates the quality of each model, relative to each of the other models given a collection of models for the data. Thus, AIC provides a means for model selection. We set random seed number to 1 and the ridge parameter to 1.0E-8 by eliminating the collinear attributes.

3.2. Features Selection

In our feature vector space, the textual features have high dimensionality though we selected only those terms having frequency equal to or more than 50 (see Section 3.3.1). Some features still may exist that are not valuable; therefore, using effective feature selection techniques improve the performance of a classifier. In this paper, we used several feature selection techniques for comparative study and their impact on the performance of classifiers. Section 4 highlights results comparison before and after feature selection techniques.

3.2.1. Particle swarm optimisation

PSO explores the feature space using the particle swarm optimisation (PSO) algorithm [30]. PSO optimizes problem stochastically by iteratively trying to improve a candidate solution with a given measure.

3.2.2. CuckooSearch

CuckooSearch (CS) feature selection technique explores the attribute space using the CS; a nature-inspired metaheuristic algorithm [31]. CS has two phases: exploration and exploitation. Exploration phase is carried out via Levy flights, whereas replacement of a fraction of attributes is performed at exploitation phase.

3.2.3. Greedystepwise

Greedy forward or backward search is carried out in the vector space. It stops when the addition/deletion of any remaining attributes results in a decrease evaluation. Greedystepwise (GW) produces a ranked list of attributes by traversing the space from one side to another and recording the order that attributes are selected [32].

3.2.4. ChiSquared

It evaluates the worth of an attribute by computing the value of ChiSquared statistical value with respect to the class [33].

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} = N \sum_{i=1}^{n} \frac{(O_i/N - p_i)^2}{p_i}, \tag{2}$$

where χ^2 is Pearson's cumulative test statistic, O_i is the number of observations of type i. N is the total number of observations, $E_i = Np_i$ = the expected (theoretical) frequency of type i, asserted by the null hypothesis that the fraction of type i in the population is p_i

3.3. Experimental Setup

We use the real-life dataset of Yelp.com crawled by Mukherjee *et al.* [12] from restaurant category and Ott *et al.* [6, 7] semi-real dataset for experiments. Ott *et al.* crawled 400 genuine reviews from tripadvisor.com and 400 spam reviews tailored by hiring Mechanical Turkers in positive polarity (Pos-Pol) [6]; whereas, for negative polarity (Neg-Pol) crawled 400 genuine reviews from expedia.com, hotels.com, orbitz.com, priceline.com, tripadvisor.com, and 400 spam reviews by Mechanical Turkers [7]. Table 1 shows the statistics of the dataset.

3.3.1. Data preprocessing and features extraction

We extract linguistic and structural features (SFs) as shown in Table 2. We select review text in the first step of data preprocessing as the text is the primary attribute of people's interest, commonly used by spammers for exploitation. By applying stop words removal and porter stemming, 12337 unique terms were identified from Yelp dataset, whereas 3663 and 5327 from Myle Ott's Pos-Pol and Neg-Pol datasets. We keep the threshold of term frequency greater or equal to 50 for each term, and we finally select 1077, 141, and 127 unique terms, respectively, for Yelp and Ott's datasets. We then calculate the *TFIDF* score determined by "Equation (3)" for these terms and name them as unique term features (UTFs). Figure 3 shows this process.

Table 1 | Dataset statistics.

Dataset	Spam Reviews	Genuine Reviews	Total
Yelp (Restaurant) Mukherjee *et al.* [12]	8303	58716	67019
Ott *et al.* [6] Pos-Pol	400	400	800
Ott *et al.* [7] Neg-Pol	400	400	800

Table 2 | Features sets.

Linguistic Features		Structural Features (SF)
UTF	**CF**	
TFIDF of selected terms	No of words No of characters No of special characters (@,#,$,%,*)	Time Rating Usefulcount Coolcount Funnycount

UTF, unique term feature; CF, content feature.

Figure 3 | Data preprocessing.

$$TFIDF\ (t,\ d,\ D) = TF\ (t,\ d)\ .\ IDF\ (t,\ D) \tag{3}$$

where $TF(t, d)$ is the number of times that term t occurs in document d and

$$IDF(t, D) = \log \frac{N}{\{d \in D : t \in D\}}$$

N is the total number of documents in corpus D and $\{d \in D : t \in D\}$ number of documents where the term t exists.

Besides UTF, we also extract content features (CFs) and SFs like date of the review posted, rating, and other metadata given by reviewers to that specific review.

3.3.2. Data normalization

We scale feature values between the range of [0, 1] by the normalization method described in "Equation (4)" due to huge variation between them.

$$a' = \frac{a - a_{min}}{a_{max} - a_{min}}, \tag{4}$$

where a' is the normalized score of each value.

3.3.3. Experimental design

In our experiments, we select randomly 2000 spam and 2000 genuine reviews from Restaurant category of Yelp.com dataset, however, used full semi-real dataset of Ott *et al.* [6, 7]. To create and examine results, we use WEKA 3.8 tool and MATLAB R2016a. We use 10-fold cross-validation for training and testing in our experiments, that is, for each run the data is randomly split into 10-folds, one of which is used as test data while the remaining 9-folds are used for training to reduce the biases of classifiers. We consider Precision, Recall, F-Measure, and Receiver Operating Characteristic (RoC) as the performance metrics.

To assert our hypothesis 1 and evaluate the performance of single classifiers and our proposed ELM; in the experiment setting one we use the following steps:

1. *Select reviews*
2. *Data preprocessing and features extraction*
3. *Use full feature sets extracted from datasets mentioned in* Table 1.
4. *Split the dataset into training and testing (90% and 10% respectively) using 10-fold cross-validation*
5. *Train the classifier using a training set*
6. *Evaluate the trained model using testing set*
7. *Store the classification performance*

To assess the performance of our proposed ELM and single classifiers with different feature selection techniques and to affirm our hypothesis 2, we adopt the following steps in setting two of our experiments:

1. *Select the reviews*
2. *Data preprocessing and features extraction*
3. *Selection of best features using feature selection technique*
4. *Add selected features in list* F_i
5. *Split the dataset into training and testing (90% and 10%, respectively) using 10-fold cross-validation*
6. *Train the classifier using a training set*
7. *Evaluate the trained model using testing set*
8. *Store the classification performance*
9. *Repeat steps 3–8 for remaining feature selection techniques.*

4. RESULTS AND DISCUSSIONS

This section discusses results derived by applying the techniques explained earlier in the Methodology section.

4.1. Performance Evaluation Using Full Feature Sets

In the first step of our experiment, we used full feature sets extracted from the datasets mentioned in Table 1. The comparative performance of base classifiers and our proposed ELM for Yelp dataset is shown in Table 3. MLP gets 0.744 precision, 0.739 recall with 0.738 F-measure and RoC 0.815, NB 0.749 precision, 0.742 recall with 0.741 F-measure and RoC 0.817, Adaboost 0.812 precision, 0.781 recall with 0.775 F-measure and RoC 0.847, whereas our proposed ELM outperforms all others by 0.842 precision, 0.834 recall 0.832 F-measure and 0.908 RoC. Our proposed ELM also outperforms base classifiers for Ott Pos-Pol and Neg-Pol datasets with 0.821 precision, 0.820 recall, 0.820 F-measure, RoC 0.888 and 0.756 precision, 0.754 recall, 0.753 F-measure, RoC 0.838, respectively, which can be seen in Tables 4 and 5. These results ascertain our hypothesis 1 which means that single classifiers have inconsistent and squat performance on redundant and low-level features.

Table 3 Results of different classifiers on yelp restaurant dataset using full (1077) features set.

Metric	MLP	NB	Adaboost	ELM
Precision	0.744	0.749	0.812	**0.842**
Recall	0.739	0.742	0.781	**0.834**
F-measure	0.738	0.741	0.775	**0.832**
RoC	0.815	0.817	0.847	**0.908**

MLP, multilayer perceptron; NB, Naïve Bayes; ELM, ensemble learning module; RoC, receiver operating characteristic.

Table 4 Results of different classifiers on M.Ott Pos-Pol dataset using full (141) features set.

Metric	MLP	NB	Adaboost	ELM
Precision	0.741	0.80	0.772	**0.821**
Recall	0.741	0.793	0.77	**0.820**
F-measure	0.741	0.791	0.77	**0.820**
RoC	0.828	0.85	0.830	**0.888**

MLP, multilayer perceptron; NB, Naïve Bayes; ELM, ensemble learning module; RoC, receiver operating characteristic.

Table 5 Results of different classifiers on Ott Neg-Pol dataset using (127) full features set.

Metric	MLP	NB	Adaboost	ELM
Precision	0.74	0.737	0.681	**0.756**
Recall	0.74	0.735	0.68	**0.754**
F-measure	0.74	0.734	0.68	**0.753**
RoC	0.798	0.808	0.772	**0.838**

MLP, multilayer perceptron; NB, Naïve Bayes; ELM, ensemble learning module; RoC, receiver operating characteristic.

4.2. Performance Evaluation Using Feature Selection Techniques

A robust system should have high performance with low processing cost and time. Adequate feature selection techniques can achieve this objective as they discard low level and redundant features. In the second strategy, we employed several features selection techniques like PSO, CS, Greedystepwise (GW), and ChiSquared to validate the comparative performance of base classifiers and our proposed ELM with these techniques. By applying PSO feature selection technique, the number of Yelp dataset attributes reduced from 1077 to 333, for M.Ott Pos-Pol dataset reduced from 141 to 21 and from 127 to 30 for M.Ott Neg-Pol dataset, respectively. CS feature selection technique reduced Yelp dataset attributes from 1077 to 510, for M.Ott Pos-Pol dataset 141 to 23 and 127 to 28 for M.Ott Neg-Pol dataset. In GW attributes of Yelp dataset reduced from 1077 to 34, M.Ott Pos-Pol dataset from 141 to 22 and 127 to 18 for M.Ott Neg-Pol dataset, respectively. We selected 25% top-ranked attributes by ChiSquared selection technique for our experiments; for Yelp dataset the number of top-ranked attributes are 270; 35 and 28 for Ott Pos-Pol and Neg-Pol datasets, respectively. Table 6 depicts the results of different feature selection techniques with different classifiers on three different datasets. It is worthy to note that for Yelp dataset, the proposed ELM performed consistently well for all feature selection techniques. NB performed slightly well for M.Ott Pos-Pol dataset for features selection CS and GW, whereas ELM outperformed others for PSO and ChiSquared techniques. MLP and NB performed well for PSO and GW techniques for Ott Neg-Pol dataset, however, ELM outperformed others for CS and ChiSquared techniques. Figures 4–6 shows the average metrics performance of base classifiers and ELM for each used dataset, where ELM outperformed all the base classifiers hence, our hypothesis 2 is also accepted.

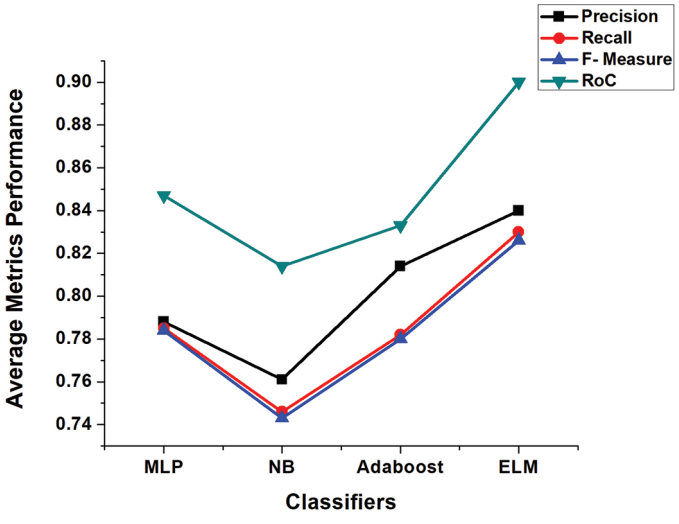

Figure 4 Average metrics performance of classifiers after feature selection techniques for Yelp dataset.

Table 6 Results of different feature selection techniques with different classifiers on different datasets.

Metric	Yelp Restaurant Dataset					Ott Pos-Pol Dataset					Ott Neg-Pol Dataset				
	FS / No of Fea	MLP	NB	Ada	ELM	FS / No of Fea	MLP	NB	Ada	ELM	FS / No of Fea	MLP	NB	Ada	ELM
Precision	PSO / 333	0.777	0.724	0.812	**0.839**	PSO / 21	0.744	0.759	0.746	**0.758**	PSO / 30	**0.752**	0.726	0.731	0.748
Recall		0.777	0.724	0.781	**0.828**		0.743	0.75	0.745	**0.754**		**0.744**	0.724	0.725	0.741
F-measure		0.777	0.723	0.775	**0.826**		0.742	0.748	0.745	**0.753**		**0.742**	0.723	0.723	0.74
RoC		0.833	0.76	0.847	**0.894**		0.812	0.83	0.806	**0.834**		0.797	0.811	0.78	**0.812**
Precision	CS / 510	0.756	0.732	0.824	**0.83**	CS / 23	0.761	**0.779**	0.754	0.767	CS / 28	0.717	0.734	0.723	**0.741**
Recall		0.755	0.732	0.773	**0.804**		0.76	**0.769**	0.753	0.763		0.709	0.734	0.716	**0.734**
F-measure		0.754	0.731	0.763	**0.8**		0.76	**0.767**	0.752	0.762		0.706	0.734	0.714	**0.734**
RoC		0.829	0.766	0.832	**0.868**		0.834	**0.852**	0.827	0.851		0.766	0.81	0.787	**0.813**
Precision	GW / 34	0.822	0.808	0.821	**0.827**	GW / 22	0.749	**0.768**	0.739	0.763	GW / 18	0.736	**0.736**	0.724	0.736
Recall		0.812	0.761	0.785	**0.811**		0.743	**0.755**	0.736	0.758		0.721	**0.731**	0.715	0.726
F-measure		0.811	0.751	0.779	**0.809**		0.741	**0.752**	0.735	0.756		0.717	**0.73**	0.712	0.723
RoC		0.88	0.873	0.794	**0.881**		0.822	**0.849**	0.815	0.845		0.786	**0.801**	0.777	0.799
Precision	ChiSq / 270	0.798	0.782	0.812	**0.851**	ChiSq / 35	0.784	0.796	0.777	**0.82**	ChiSq / 32	0.74	0.761	0.704	**0.774**
Recall		0.798	0.77	0.781	**0.842**		0.784	0.783	0.776	**0.818**		0.73	0.76	0.704	**0.764**
F-measure		0.797	0.768	0.775	**0.841**		0.784	0.78	0.776	**0.817**		0.727	0.76	0.704	**0.761**
RoC		0.846	0.858	0.847	**0.918**		0.867	0.886	0.84	**0.891**		0.802	0.84	0.795	**0.842**

FS, feature selection technique; Fea, features; Ada, Adaboost; ELM, ensemble learning module; CS, CuckooSearch; GW, Greedystepwise; RoC, receiver operating characteristic.

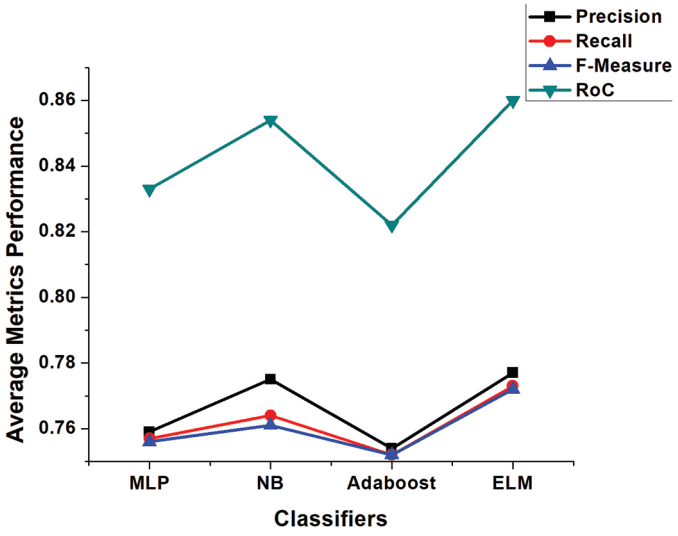

Figure 5 Average metrics performance of classifiers after feature selection techniques for M.Ott Pos-Pol.

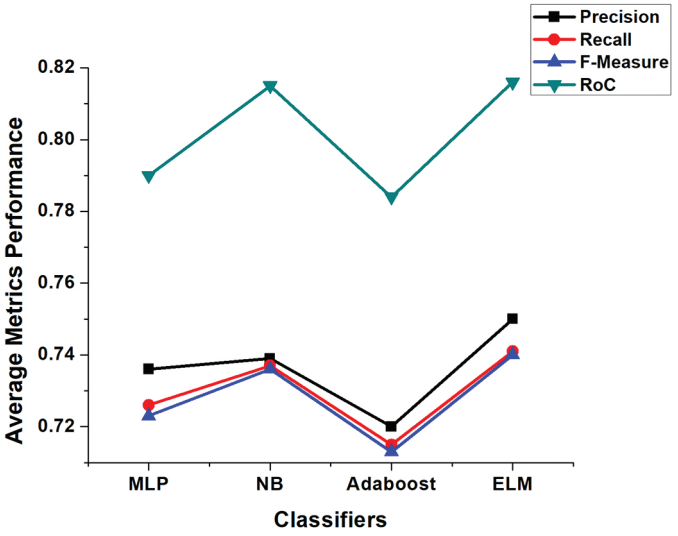

Figure 6 Average metrics performance of classifiers after feature selection techniques for M.Ott Neg-Pol dataset.

5. CONCLUSION

In this study, we investigated the performance and effectiveness of ensemble learning with feature selection techniques for review spam detection on real and semi-real-life datasets. We carried out experiments in two settings, where setting one uses full feature sets; however, different feature selection techniques have been employed in setting two to remove redundant and low-level features. Our proposed ELM outperformed base classifiers in both settings. It is observed that ChiSquared feature selection technique with ELM has outperformed other feature selection techniques and base classifiers with Precision (0.851, 0.820, and 0.774) for Yelp, M.Ott Pos-Pol, and M.Ott Neg-Pol datasets, respectively.

The limitation of this study is the use of imbalance dataset because practically, spams are always less in number than genuine reviews resulting in imbalance dataset. In the future, we will focus on this data imbalance issue. We will also try to reduce the cost impact of misclassifications.

ACKNOWLEDGMENTS

This work is supported by the Academic and Technological Leadership Training Foundation of Sichuan Province, China (WZ0100112371601/004, WZ0100112371408, YH1500411031402).

REFERENCES

[1] A.K. Samha, Y. Li, J. Zhang, Aspect-based opinion extraction from customer reviews, Computer Science and Information Technology (CS and IT), Volume 4, Number 4, in Proceedings of the Second International Conference of Database and Data Mining (DBDM 2014), Dubai, 2014, pp. 149–160.

[2] M. Luca, G. Zervas. Fake it till you make it: reputation, competition, and Yelp review fraud. Manag. Sci. 62(12) (2016), 3412–3427.

[3] N. Jindal, B. Liu, Analyzing and detecting review spam, in Seventh IEEE International Conference on Data Mining (ICDM), Omaha, 2007, pp. 547–552.
[4] N. Jindal, B. Liu, Opinion spam and analysis, in Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM'08), Palo Alto, 2008, pp. 219–230.
[5] D.H. Fusilier, M. Montes-y-Gómez, P. Rosso, R.G. Cabrera. Detecting positive and negative deceptive opinions using PU-learning. Info. Proc. Manag. 51(4) (2015), 433–443.
[6] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Portland, 2011, pp. 309–319.
[7] M. Ott, C. Cardie, J.T. Hancock, Negative deceptive opinion spam, in The 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013), Atlanta, 2013, pp. 497–501.
[8] S. Feng, R. Banerjee, Y. Choi, Syntactic stylometry for deception detection, in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Jeju Island, 2012, pp. 171–175.
[9] J. Li, C. Cardie, S. Li, TopicSpam: a Topic-Model based approach for spam detection, Assoc. Comput. Linguist. 2 (2013), 217–221.
[10] D.H. Fusilier, M. Montes-y-Gómez, P. Rosso, R.G. Cabrera, Detection of opinion spam with character n-grams, in: Computational Linguistics and Intelligent Text Processing, Springer, 2015, pp. 285–294.
[11] F. Khurshid, Y. Zhu, C. W. Yohannese, M. Iqbal, Recital of supervised learning on review spam detection: an empirical analysis, in 2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing, 2017, pp. 224–229.
[12] A. Mukherjee, V. Venkataraman, B. Liu, N.S. Glance, What yelp fake review filter might be doing?, in The seventh International AAAI Conference on Weblogs and Social Media (ICWSM), Massachusetts, 2013, pp. 409–418.
[13] M. Crawford, T.M. Khoshgoftaar, J.D. Prusa, A.N. Richter, H. AlNajada. Survey of review spam detection using machine learning tecnoques. J. Big Data. 2 (2015), 1–24.
[14] J. Li, M. Ott, C. Cardie, E.H. Hovy, Towards a general rule for identifying deceptive opinion spam, in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Baltimore, 2014, pp. 1566–1576.
[15] T. Wang, H. Zhu, Voting for deceptive opinion spam detection, arXiv preprint arXiv: 1409.4504, 2014.
[16] S. Banerjee, A. Chua, A linguistic framework to distinguish between genuine and deceptive online reviews, in Proceedings of the International Multi Conference of Engineers and Computer Scientists IMECS, Hong Kong, 2014, Vol I, pp. 501–506.
[17] A. Mukherjee, V. Venkataraman, B. Liu, N. Glance, Fake review detection: classification and analysis of real and pseudo-reviews, Technical Report UIC-CS-2013-03, University of Illinois, Chicago, 2013.
[18] C. Chen, H. Zhao, Y. Yang, Deceptive opinion spam detection using deep level linguistic features, in Natural Language Processing and Chinese Computing - 4th CCF Conference, NLPCC 2015, Springer, Nanchang, Oct. 9–13, 2015, pp. 465–474.
[19] Y. Ren, D. Ji. Neural networks for deceptive opinion spam detection: an empirical study. Info. Sci. 385 (2017), 213–224.
[20] S. Xie, G. Wang, S. Lin, P.S. Yu, Review spam detection via temporal pattern discovery, in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, 2012, pp. 823–831.
[21] N. Jindal, B. Liu, E.-P. Lim, Finding unusual review patterns using unexpected rules, in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, 2010, pp. 1549–1552.
[22] H. Li, L. Bing, A. Mukherjee, J. Shao. Spotting fake reviews using positive-unlabeled learning. Comput. Syst. 18(3) (2014), 467–475.
[23] H. Li, Z. Chen, L. Bing, X. Wei, J. Shao, Spotting fake reviews via collective positive-unlabeled learning, in Data Mining (ICDM), 2014 IEEE International Conference on, 2014, pp. 899–904: IEEE.
[24] G. John, P. Langley, Estimating continuous distributions in Bayesian classifiers, in Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, 1995, pp. 338–345.
[25] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in Thirteenth International Conference on Machine Learning, San Francisco, 1996, pp. 148–156.
[26] J. Su, H. Zhang, C.X. Ling, S. Matwin, Discriminative parameter learning for Bayesian networks, in The 25th International Conference on Machine Learning (ICML 2008), Helsinki, 2008, pp. 1016–1023.
[27] F. Ahmed, M. Abulaish, A generic statistical approach for spam detection in online social networks. Comput. Commun. 36(10) (2013), 1120–1129.
[28] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2(3) (2011), pp. 1–27.
[29] N. Sugiura, Further analysis of the data by Akaike's information criterion and the finite corrections, Commun. Stat. Theory Methods. 7(1) (1978), 13–26.
[30] A. Moraglio, C. DiChio, R. Poli, Geometric Particle Swarm Optimisation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 125–136.
[31] X.S. Yang, D. Suash, Cuckoo search via Lévy flights, in 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, 2009, pp. 210–214.
[32] P.E. Black, Greedy algorithm, in: V. Pieterse, P.E. Black (Eds.), Dictionary of Algorithms and Data Structures, Feb. 2, 2005.
[33] K. Pearson, X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling AU, Lond. Edinb. Dublin Philos. Mag. J. Sci. 50 (July 1, 1900), 157–175.