Received December 28, 2016, accepted January 13, 2017, date of publication January 18, 2017, date of current version March 13, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2655032

Revisiting Semi-Supervised Learning for Online Deceptive Review Detection

JITENDRA KUMAR ROUT[1], ANMOL DALMIA[1], KIM-KWANG RAYMOND CHOO[2], (Senior Member, IEEE), SAMBIT BAKSHI[1], (Member, IEEE), AND SANJAY KUMAR JENA[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, National Institute of Technology Rourkela, Odisha 769 008, India
[2]Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249- 0631, USA

Corresponding author: S. Bakshi (sambitbaksi@gmail.com)

This work was supported in part by the Information Security Education and Awareness Project (Phase II), Ministry of Electronics and Information Technology (MeitY), Government of India, and in part by the Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions Program 2016, Department of Science and Technology, Government of India, under Grant ETI/359/2014.

ABSTRACT With more consumers using online opinion reviews to inform their service decision making, opinion reviews have an economical impact on the bottom line of businesses. Unsurprisingly, opportunistic individuals or groups have attempted to abuse or manipulate online opinion reviews (e.g., spam reviews) to make profits and so on, and that detecting deceptive and fake opinion reviews is a topic of ongoing research interest. In this paper, we explain how semi-supervised learning methods can be used to detect spam reviews, prior to demonstrating its utility using a data set of hotel reviews.

INDEX TERMS Online review spam, semi-supervised learning, unlabeled reviews, PU learning, Co-training, EM algorithm, label propagation and spreading.

I. INTRODUCTION

Opinion spamming is becoming more sophisticated and, in some cases, organized, due to the potential to profit from such activities. For example, some businesses reportedly recruited online users (e.g. professional fake review writers) to post fake opinions. These opinions can be used to market and promote a particular business, spread rumors and damage the reputation of a competing business, or influence online users' opinions and views about a particular topic (e.g. during elections) [1].

Unlike other forms of spam [2]–[4], it is challenging to identify fake opinions, as one may need to also understand the context of the postings in order to determine whether the particular opinion is deceptive [5]–[7]. For example, how can one reliably determine whether the online review postings about a particular business (e.g. reviews about a restaurant) reflect the actual experience of the users who had posted the reviews? One could, perhaps, examine the online review posting history of these users and make a determination whether a particular user is posting multiple (near) duplicate reviews about different businesses in a particular time frame. However, the latter scenario may only be a small percentage of deceptive online opinions from review sites [8]. Also, how does one determine whether postings about a particular politician accurately reflect the feelings of the electorate, and the postings are not written by a single individual or group of individual purporting to be different persons – a practice also known as astroturfing [1], [6], [7])?

While supervised learning has been traditionally used to detect fake reviews [9]–[15], supervised learning approaches suffer from several limitations. For example, unless one can be assured of the "quality" of the reviews used in the training dataset, we will have a garbage-in-garbage-out situation. In addition, the amount of labeled data points used to train the classifier can be difficult to obtain and update, given the dynamic nature of online reviews. In [14], the authors highlighted that human are poor in labeling reviews as fake or genuine. This complicates the task of finding ground truth for given instances accurately. Due to lack of reliable data and the dynamic nature of online reviews, unsupervised methods (see [16]–[19]) or methods based on heuristic rules (see [12], [20]–[22]) have also been used to detect deceptive online reviews.

Some limitations in supervised learning methods could be addressed using automatic labeling, a process known as semi-supervised learning. In the latter, a large number of unlabeled data points are used, instead of labeled data points. As such, labeled data points can be sparsely present and using those points, labels of the unknown instances are automatically generated first, which can then be used to train a classifier and evaluate a given review.

Thus, in this paper, we use several semi-supervised learning approaches to improve the classification, as well as incorporating three new dimensions in the feature vector (i.e. Parts-of-Speech features, Linguistic and Word Count features and Sentimental Content features) to obtain better results. We then evaluate the proposed approach using a dataset comprising both positive and negative reviews.

In the next section, we review and analyze related work. Section III-A describes the proposed approach and experiment setup. We present and discuss our findings in Section IV, before concluding the paper in Section V.

II. RELATED WORK

Deceptive online review detection is generally considered a classification problem [9], [14], and one popular approach is to use supervised text classification techniques [9], [10], [13]–[15]. These techniques are robust if the training is performed using large datasets of labeled instances from both classes, deceptive opinions (positive instances) and truthful opinions (negative examples). However, it is challenging in practice to obtain such large and accurate training sets. Ott *et al.* [14] explained that identification of deceptive online reviews is often performed using prior human knowledge, which increases the probability of mislabeled reviews due to the potential for subjectivity during the labeling process.

Therefore, in studies such as those of [14], [23]–[25], synthetic datasets of deceptive reviews are used. In these approaches, the classification of reviews is performed by investigating the psycholinguistic and structural differences between deceptive and non-deceptive reviews.

It is also easier to collect a large amount of unlabeled reviews, in comparison to labeled reviews required in the training of supervised techniques. Thus, if we have a large number of unlabeled reviews, a viable approach is to use semi-supervised techniques. For example, Li *et al.* [26] used review and reviewer features to design a two-view semi-supervised method, by employing the framework of co-training algorithm [27] to detect spam reviews. In this approach, the co-training algorithm uses the large amount of unlabeled examples to train the algorithm [28]. More recently in 2016, Zhang *et al.* [29] presented a co-training approach, Co-training for Spam review identification (CoSpa), to identify spam reviews. In the approach, spam reviews are identified by two views, namely: the set of lexical terms derived from the textual content of the reviews and the set of Probabilistic Context-Free Grammars (PCFG) rules derived from a deep syntax analysis of the reviews. Using Support Vector Machine (SVM) as the base classifier, the authors developed two strategies, namely: CoSpa-C and CoSpa-U. Experimental results demonstrated that both variants outperform the SVM classifier when applied on PCFG rules on lexical tokens. The work in [30] introduced a three-view semi-supervised method, tri-training, which uses labeled data. The annotated data is increased by adding unlabeled data incrementally in a feedback fashion. In the context of deceptive review spam identification, each given review has three types of features, namely: *review features*, *reviewer features*, and *store features*.

However, the use of co-training in classification suffers from several drawbacks. For example, the manually labeled reviews used in co-training can be unreliable due to human involvement and subjectivity (e.g. [14] reported only a 60% accuracy rate). The use of only positive and unlabeled data leads to poor performance in co-training algorithms [31]. Such approach also does not consider the features of deep syntax and psychological linguistics of review text, which can help improve the effectiveness of deceptive review detection. Thus, in this paper, a two-view co-training approach using these features is proposed.

Positive Unlabeled (PU) learning is another semi-supervised learning approach [32], [33], which can be used to build an accurate classifier even without having labeled negative training examples. Several PU learning techniques have been applied successfully in document classification with promising results [34]–[37]. Hernández *et al.* [38] first used this technique to detect review spam. Specifically, the authors proposed PU-LEA, which adapts the PU-learning approach [32], [33]. PU-learning reportedly achieves an F-measure of 83.7% with Näive Bayes, using only 100 positive examples. While this is better than the findings reported by Li *et al.* [26], where 6000 labeled instances and co-training were used, it is difficult to make a conclusive statement as both approaches use different datasets. The dataset of Ott *et al.* [14] may not provide an accurate indication of real world performance [39]. Also, their assumption regarding continual refining of negative instances over iterations will not always hold in practice, as pointed out by Li *et al.* [40]. Li *et al.* [41] then showed that PU-LEA identified much fewer positive examples from the unlabeled set. In addition, the authors attempted to detected review spams in Chinese-language reviews of restaurants from Dianping.com. In their approach, LPU [42] was used, also considering the fact that unknown set is really an unlabeled set rather than the non-fake review set. According to the authors, PU learning not only outperforms SVM but also detects a large number of potentially fake reviews hidden in the unlabeled set. The authors used publicly available PU learning system. However, data from Dianping.com were filtered and it is known that using filtered fake reviews is not as effective and efficient [43]. Moreover, the authors only used the Unigrams and Bigrams features, without considering other relevant features.

Li *et al.* [41] studied the problem of fake review detection using the Collective PU (CPU) learning framework, and they proposed a collective classification algorithm, Multi-typed Heterogeneous Collective Classification (MHCC), designed to work in a heterogeneous network of reviews, users and IPs. The authors reported that their approach not only outperforms

TABLE 1. Summary of semi-supervised algorithms.

Authors	Approach	Key Concept
Blum *et al.* [27]	Combine labeled and unlabeled data using co-training	Independence of feature components
Zhu *et al.* [47]	Propagation of labels for labeling unlabeled instances	Graphical structure of the feature vector space
Zhu *et al.* [48]	Spreading of labels using spectral functions	Spectral properties of the feature vector space
Karimpour *et al.* [46]	Expectation Maximization to generate a classifier	Iteratively identify correct labels of unlabeled data
Hernández *et al.* [38]	Labeling unlabeled data using positively labeled examples	Iteratively identify correct positively labeled data from unlabeled data

baseline approaches but, more importantly, detects a large number of potential fake reviews hidden in the unlabeled set. It was also reported that the models use language independent features, and hence they can be generalized to any other languages.

Ren *et al.* [44] proposed the Mixing Population and Individual Property PU Learning (MPIPUL) model, which is designed to deal with easily mislabeled (spy) examples in unlabeled reviews not addressed by previous techniques. The process begins with the identification of some reliable negative examples from the unlabeled dataset, followed by the generation of some representative positive examples and negative examples using Latent Dirichlet Allocation (LDA). Then, the remaining spy examples that cannot be explicitly identified as positive or negative are assigned two similarity weights. The weights are used to evaluate their probabilities and determine whether they belong to the positive or negative class. Finally, spy examples and their similarity weights are incorporated into a SVM classifier.

Hernández *et al.* [45] attempted to detect both positive and negative deceptive reviews, by taking a more conservative approach than the original PU-learning approach. Specifically, the authors selected reliable negative examples (i.e., genuine reviews) from unlabeled ones as well as analyzing the role of opinion polarity. Their evaluations found that the proposed PU-learning method consistently outperformed the original PU-learning approach, with an average improvement of 8.2% and 1.6% over the original approach in the detection of positive and negative deceptive opinions respectively.

In this paper, we present a comprehensive approach with extended feature sets using five popular semi-supervised learning techniques, in order to support larger and varied datasets.

III. PROPOSED APPROACH AND EXPERIMENT SETUP

A. SEMI-SUPERVISED METHODS ADAPTED FOR REVIEW SPAM DETECTION

The following feature points were chosen to be extracted and used for the experiments from the dataset:

- Sentiment Polarity
- Parts of Speech (POS) tags
- Linguistic Inquiry and Word Count (LIWC)
- Bigram frequency counts

In this paper, we implemented and evaluated four different state-of-the-art semi-supervised learning models (see Sections III-A1 to III-A4). Table 1 summarizes the algorithms evaluated in this article for review spam detection.

1) CO-TRAINING ALGORITHM

Co-training is a method which allows the combination of labeled and unlabeled instances to form a labeled training dataset. This method is primarily based on a PAC-style learning algorithm proposed by Blum *et al.* [27]. The method is originally deployed for classifying web spam data, and assumes each example in the dataset to consist of two views of data. Each view is a distribution of features that make up the example. The idea is to train two classifiers on each view and then classify instances on the unlabeled category to enlarge the training set. The condition here is that the two views should not be directly co-relatable with each other.

Initially, a collection of data points is chosen, of which some are labeled (L) and the others are unlabeled (U). The U set is then iteratively exhausted by incrementally learning and classifying member instances to the L set. First, u instances are considered at random from U and inserted into a set U'. Each instance is a composition of two views, x_1 and x_2. The algorithm then runs for k iterations or until the set U is exhausted. In each iteration, a classifier h_1 is trained on only the x_1's view of the instances in L, and another classifier h_2 on only the x_2's view of the instances in L. Then, each classifier is allowed to label p positive and n negative instances, which are added to the set L. Finally, $2(p+n)$ examples are randomly sampled from U and are used to replenish U'.

The co-training algorithm is described in Algorithm 1.

2) EXPECTATION MAXIMIZATION ALGORITHM

The Expectation Maximization algorithm, first proposed by Karimpour *et al.* [46], is designed to label unlabeled data to be used for training. The algorithm operates as follows: a classifier is first derived from the labeled dataset. This classifier is then used to label the unlabeled dataset. Let this predicted set of labels be PU. Now, another classifier is derived from the combined sets of both labeled and unlabeled datasets and is used to classify the unlabeled dataset again. This process is repeated until the set PU stabilizes. After a stable PU set is produced, we learn the classification algorithm with the combined training set of both labeled and unlabeled datasets and deploy it for predicting test dataset.

Here, the learning of the algorithm with the conjunction of the labeled and predicted labeled sets is the Expectation step (E-step) and the prediction of the labels of the unlabeled set is the Maximization step (M-step). The pseudocode for EM learning is described in Algorithm 2.

Algorithm 1 Co-Training Algorithm

INPUT: Labeled instance set L, and unlabeled instance set U.
OUTPUT: Deployable classifier, C.

Create set of unlabeled examples, U', by randomly sampling u examples from U;
for each feature vector x in $L \cup U$ **do**
  partition x to tuple of views, (x_1, x_2);
end for
for k iterations **do**
  $h_1 \leftarrow train(x_1)\ \forall (x_1, x_2) \in L$;
  $h_2 \leftarrow train(x_2)\ \forall (x_1, x_2) \in L$;
  Let h_1 label p positive and n negative examples from U';
  Let h_2 label p positive and n negative examples from U';
  Add labeled examples to L;
  Randomly sample $2(p + n)$ examples from U to U';
end for

Algorithm 2 EM Algorithm

INPUT: Labeled instance set L, and unlabeled instance set U.
OUTPUT: Deployable classifier, C.

$C \leftarrow train(L)$;
$PU = \emptyset$;
while *true* **do**
  $PU = predict(C, U)$;
  if PU same as in previous iteration **then**
    return C;
  end if
  $C \leftarrow train(L \cup PU)$;
end while

3) LABEL PROPAGATION AND SPREADING

Label propagation is first proposed for semi-supervised learning by Zhu *et al.* [47]. In this model, the learning algorithm is a graph-based algorithm, where each node stores some information about its label. The graph is constructed by ordering suitable vector feature based on a suitable similarity metric, such as Manhattan distance or Euclidean distance. Each node can be either labeled or unlabeled. In the process, label information is broadcasted across the graph dynamically and finally, all nodes are labeled.

Label propagation is useful when definite values are available that can give a meaningful ordering of the data instances. However, in practice, such data points are either faulty or incomplete. For example, many real-world feature vectors have missing data entries. Label propagation, as such, is less useful. To overcome this problem, label spreading algorithm is used that allows soft clamping of data and finding spectral clusters in the graph, and hence is more resistant to noise than label propagation (refer to [48]).

TABLE 2. Performance metrics for co-training based approach.

Partition	Learner	Accuracy	Precision	Recall	F-Score
75-25	k-NN	0.7650	0.8100	0.7431	**0.7751**
	Logistic Regression	0.5025	0.9950	0.5012	0.6667
	Random Forest	0.6075	0.7800	0.5799	0.6652
	Stochastic Gradient Descent	0.5075	0.9900	0.5038	0.6678
80-20	k-NN	0.7469	0.8063	0.7207	0.7612
	Logistic Regression	0.5094	1.0000	0.5047	0.6709
	Random Forest	0.7281	0.9000	0.6697	0.7680
	Stochastic Gradient Descent	0.5031	1.0000	0.5016	0.6681
90-10	k-NN	0.7375	0.8750	0.6863	0.7692
	Logistic Regression	0.5125	1.0000	0.5063	0.6723
	Random Forest	0.6813	0.8000	0.6465	0.7151
	Stochastic Gradient Descent	0.6750	0.9500	0.6129	0.7451

TABLE 3. Performance metrics for EM algorithm based approach.

Partition	Learner	Accuracy	Precision	Recall	F-Score
75-25	k-NN	0.8300	0.8500	0.8173	**0.8333**
	Logistic Regression	0.8250	0.8000	0.8421	0.8205
	Random Forest	0.7450	0.7050	0.7663	0.7344
	Stochastic Gradient Descent	0.5475	0.9900	0.5252	0.6863
80-20	k-NN	0.8313	0.8063	0.8487	0.8269
	Logistic Regression	0.8281	0.7750	0.8671	0.8185
	Random Forest	0.7094	0.6125	0.7597	0.6782
	Stochastic Gradient Descent	0.7000	0.9750	0.6290	0.7647
90-10	k-NN	0.8000	0.8250	0.7857	0.8049
	Logistic Regression	0.8000	0.8375	0.7791	0.8072
	Random Forest	0.7500	0.7625	0.7439	0.7531
	Stochastic Gradient Descent	0.7625	0.9875	0.6810	0.8061

4) POSITIVE UNLABELED LEARNING

PU learning generates a two-class classifier based on positively labeled or unlabeled examples. The uniqueness of this approach is its ability to identify hidden positives from the set of unlabeled examples when negative training data is not supplied or available. PU learning has two variations based on the usage of unlabeled data in the process. Both variations make use of positive examples to produce the final classifier. While one family of methods utilizes only a few examples from the unlabeled set [32], [33], [49], the other generates classifier uses the entire unlabeled dataset [36], [37].

Hernández *et al.* [38] applied this algorithm for deceptive review detection using half the datasets used here. Although [38] achieved an F-Score of 0.837 using just 100 positive instances for training, the results published did not disclose the accuracy or feature characteristics of their methods, which made it difficult to compare performance. We remark that both datasets also have different sentimental polarities.

Our approach in this paper is as follows: a set of labeled positive data points and a set of unlabeled data points are used for training. We first train a classifier with the conjunction of the positive labeled set and the existing unlabeled set. Then, this classifier is used to label the instances of the current unlabeled set. The positively labeled instances are extracted

Algorithm 3 PU Algorithm

INPUT: Positively labeled instance set P and unlabeled instance set U.

OUTPUT: Deployable classifier, C.

$i \leftarrow 1$;
$|W_0| \leftarrow |U|$;
$|W_1| \leftarrow |U|$;
while $|W_i| \leq |W_{i-1}|$ **do**
  $C_i \leftarrow train(P, U_i)$;
  $U_i^L \leftarrow predict(C_i, U_i)$;
  $W_i \leftarrow extract\_positives(U_i^L)$;
  $U_{i+1} \leftarrow U_i - W_i$;
  $i \leftarrow i + 1$;
end while
return C_i;

TABLE 4. Performance metrics for label-propagation approach.

Partition	Learner	Accuracy	Precision	Recall	F-Score
75-25	*k*-NN Kernel	0.8250	0.8350	0.8186	**0.8267**
80-20	*k*-NN Kernel	0.8313	0.7938	0.8581	0.8247
90-10	*k*-NN Kernel	0.8000	0.8125	0.7927	0.8025

TABLE 5. Performance metrics for label-spreading approach.

Partition	Learner	Accuracy	Precision	Recall	F-Score
75-25	*k*-NN Kernel	0.8275	0.8050	0.8429	**0.8235**
80-20	*k*-NN Kernel	0.8313	0.7813	0.8681	0.8224
90-10	*k*-NN Kernel	0.8125	0.7750	0.8378	0.8052

from the completed labeling. After the extraction, the next unlabeled set is created by removing the extracted positives from the current unlabeled set. This process is repeated until the current unlabeled set becomes smaller in size than its previously generated unlabeled set. After this loop is terminated, the classifier obtained in the last iteration is returned for classification purposes. This process not only labels the unlabeled dataset, but also incrementally develops the final classifier.

The PU learning is described in Algorithm 3.

B. DATASET DESCRIPTION

In this paper, the 'gold standard' dataset by Ott *et al.* [14], [50] is used in our evaluations. The dataset comprises 1,600 review texts on 20 hotels in the Chicago area, USA, which have 800 deceptive reviews and 800 genuine reviews. For the evaluations, a tag of '1' denotes deceptive reviews, highlighting that they are treated as the positive instances, whereas '0' denotes genuine reviews. In the dataset, 400 are written with a negative sentimental polarity and 400 depict a positive sentimental polarity. These reviews were obtained from various sources. The deceptive reviews were generated using Amazon Mechanical Turk (AMT) and the rest obtained from various online reviewing websites such as Yelp, TripAdvisor, Expedia, and Hotels.com.

For the evaluations, the dataset is partitioned in a fixed way. Of the 1600 examples in the corpus, two sets of examples were created, namely: the training set and the test set. The proportions partition the corpus in ratios of 75:25, 80:20, and 90:10 according to the 4-fold, 5-fold and 10-fold partitioning schemes, respectively. The examples in each set are chosen using stratified random sampling on the complete corpus such that half the examples are deceptive and half are honest in each set.

IV. FINDINGS AND DISCUSSION

As previously discussed, the available dataset was partitioned into subsets with sizes in the ratios of $a : (100 - a)$, where a assumes values in $\{75,\ 80,\ 90\}$. In each process described, $(0.2 \times a)\%$ instances were taken as labeled training dataset and the rest as unlabeled training dataset. Also, four variations of classifiers were used across all evaluations, namely the *k-Nearest Neighbor classifier* (*k*-NN), the *Logistic Regression classifier*, the *Random Forest classifier* and the *Stochastic Gradient Descent classifier*. For the *k*-NN classifier, the value of '*k*' was chosen as 4. Also, for the Random Forest classifier, 100 worker instances were used for evaluations.

The algorithms implemented and their results are presented in Sections IV-A to IV-D.

TABLE 6. Performance metrics for PU learning based approach.

Partition	Learner	Accuracy	Precision	Recall	F-Score
75-25	*k*-NN	0.7626	0.9150	0.7011	0.7939
	Logistic Regression	0.8300	0.8300	0.8300	0.8300
	Random Forest	0.5800	0.8500	0.5519	0.6693
	Stochastic Gradient Descent	0.5975	0.9950	0.5543	0.7120
80-20	*k*-NN	0.8250	0.8500	0.8095	0.8293
	Logistic Regression	0.8375	0.8313	0.8418	**0.8365**
	Random Forest	0.5969	0.8500	0.5643	0.6783
	Stochastic Gradient Descent	0.7938	0.6750	0.8852	0.7660
90-10	*k*-NN	0.7750	0.8500	0.7391	0.7907
	Logistic Regression	0.7688	0.8625	0.7263	0.7886
	Random Forest	0.6375	0.9250	0.5873	0.7184
	Stochastic Gradient Descent	0.7125	0.8750	0.6604	0.7527

A. CO-TRAINING ALGORITHM

Although Blum *et al.* [27] used two-dimensional feature vectors for web spam data for classification, the dataset used in this paper has much more sophisticated feature vectors with more than 15 dimensions. Thus, each half of a feature vector was considered as a view and the algorithm was applied as such. For the runs, the values of the parameters were set as $p = 1, n = 3, k = 30$ and $u = 75$ as derived from [27].

For the evaluations, the best score obtained was 76.50% accuracy and an F-Score of 0.775. In this particular evaluation, the dataset was divided in a 75:25 partition for training and test dataset. Of the training dataset, 20% of the instances were chosen as labeled and the rest as unlabeled. The *k*-NN classifier was used for the evaluations, and the findings are presented in Table 2.

TABLE 7. Comparative summary of semi-supervised learning techniques.

Works	Dataset(s) Used	Feature(s) Employed	Algorithm(s) Used	Performance Scores				Specifications for Optimal Results
				Accuracy	Precision	Recall	F-Score	
Zhang *et al.* [29]	Ott AMT, Web (+, -)	Word terms, Probabilistic CFG Rules	Co-training (CoSpa- U)	∼0.94	–	–	–	Uniform selection of unlabeled data in 30 iterations, $n = p = 3$
Ren *et al.* [51]	Ott AMT, Web (+)	Topic Distribution	LELC	–	0.83	–	–	40% of the truthful reviews as the training data
			SPUL-Local	–	0.84	–	–	
			SPUL-Global	–	0.85	–	–	
			MPIPUL	–	0.87	–	–	
Li *et al.* [26]	Epinion crawl, ∼60k reviews	Content, meta, product, reviewer, senti	Co-training	–	0.64	0.62	0.63	40 iterations, $p : n = 1 : 3$
Chengzhang and Kang [30]	AliExpress crawl, ∼2.3k reviews	review, reviewer, store	Tri-training	–	0.71	0.69	0.70	10-fold
Li *et al.* [40]	Yelp Dataset	Unigrams and bigrams as TF-IDF values	Spy + EM	-	0.59	0.89	0.71	MCS >= 0.8, ANR >= 2
Li *et al.* [41]	Dianping crawl, ∼9.7k reviews	Unigrams and bigrams as TF-IDF values	Collective PU	∼0.82	∼0.81	∼0.72	∼0.75	5-fold
Fusilier *et al.* [45]	Ott AMT, Web (+, -)	Unigrams and Bigrams	Modified PU	–	0.77	0.87	0.80	Results for detection in only truthful review set, 5-fold
Hernández *et al.* [38]	Ott AMT, Web (+)	–	PU	–	0.78	0.90	0.84	Results for detection in only deceptive review set, 5-fold
Proposed Work	Ott AMT, Web (+, -)	Bigrams, sentiment score, POS, LIWC	Co-training	0.77	0.81	0.74	0.78	5-fold, $p : n = 1 : 3$, 30 iterations
			Label Spreading	0.83	0.81	0.84	0.82	4-fold
			Label Propagation	0.83	0.84	0.82	0.83	
			EM	0.83	0.85	0.82	0.83	
			PU	0.84	0.83	0.84	0.84	Results for detection in consolidated set, 5-fold

B. EXPECTATION MAXIMIZATION ALGORITHM

In [46], the authors used the EM algorithm for classification of web spam data similar to [27]. The principles of application remain the same for this paper. For the evaluations, the best results were obtained when the training set was partitioned in the ratio of 75:25. An accuracy of 83% and an F-Score of 0.833 were obtained with the use of the *k*-NN classifier. The various performance indicators for the experiments using the EM algorithm are presented in Table 3.

C. LABEL PROPAGATION AND SPREADING

For Label Propagation, the best score obtained had an accuracy of 82.5% and an F-Score of 0.827. For Label Spreading, the best score with similar accuracy and an F-Score of 0.824 was obtained, which is comparable to that of Label Propagation algorithm. In both evaluations, *k*-NN kernel was used for the algorithms. Tables 4 and 5 present the evaluation findings for Label Propagation and Spreading, respectively.

D. POSITIVE UNLABELED LEARNING

The PU algorithm was implemented and evaluated using the dataset described in Section III-B. The best results were obtained when the dataset was partitioned 80% for training and 20% for testing. Out of the 80% training data comprising 1280 instances, 256 positively labeled instances were chosen as labeled instances and the remaining instances were treated as unlabeled. This is considerably close to the 200 positively labeled instances used in [45] for the same purpose. Because of additional variations in our dataset, a balanced mix of 320 data points was chosen for testing purposes as compared to 160 used in [38] and the same in [38], [45] which reported a maximum F-Score of 0.837 when applied only on the set of deceptive reviews with the mentioned dataset partitioning scheme and an undisclosed accuracy of classification. The authors in [45] reported a maximum F-Score of about 0.796 when applied on a mixed polarity training dataset like the one used in this work but having unequal numbers of deceptive and honest opinions. In our evaluations, an F-Score of 0.8365 was obtained with the partitioning scheme described, using the Logistic Regression classifier as the base classifier.

An accuracy of 83.75% was obtained, which outperforms the human accuracy reported by Ott *et al.* [14] and in [38]. We also noted that in [38], an F-Score of 0.811 was reported for the same scheme in the truthful opinion class, whereas our system reports an overall score of 0.837 when applied on a mixed set of instances consisting of balanced proportions of deceptive and genuine reviews. Performance metrics for experiments conducted for PU Learning based approach are presented in Table 6.

The performance measures of the proposed algorithm are compared to those described in Table 7, where *CFG* denotes *Context Free Grammar*, *MCS Maximum Content Similarity* and *ANR Average number of reviews per day*. Also, '–' denotes *unspecified detail*. The comparison spans across various applications of semi-supervised learning in detection of fake reviews, web spam detection, etc. The comparison emphasizes on the performance of each approach in the context of dataset used and features considered. It also compares the judgment of computational complexity and requirements, the experimental conditions that yielded the best results as mentioned in the respective literature, etc. The entries in Table 7 are ordered as per the F-score of performance.

V. CONCLUSION AND FUTURE WORK

With the increasing influence of online opinion and reviews on users, the capability to detect deceptive online reviews is crucial.

In this paper, we demonstrated how four popular semi-supervised learning approaches can be used to improve the F-score metric in classification. By incorporating new dimensions in the feature vector, namely: Parts-of-Speech features, Linguistic and Word Count features and Sentimental Content features, we obtained better results. The dataset used in our evaluations was "richer" than previously used datasets in the sense that it contains reviews with both positive and negative opinions. Using our approach, we achieved an F-score of 0.837 using PU Learning based classification. This demonstrated the usefulness of the feature vectors used in this paper.

Future research along this direction includes implementing and evaluating the proposed approach in the real-world, for example, using the approach on data collected from various websites in real-time. Also, minimal meta-data are considered in this work during classification. Future investigation may include a better integrating of minimal meta-data. Apart from textual content, associated multimedia content can also be considered for further study.

ABBREVIATIONS

ANR	:	Average Number of Reviews per day
CFG	:	Context-Free Grammar
CoSpa	:	Co-training for Spam review identification
CPU Learning	:	Collective Positive and Unlabeled Learning
EM Algorithm	:	Expectation Maximization Algorithm
LDA	:	Latent Dirichlet Allocation
LELC	:	Learning by Extracting Likely positive and negative micro-Clusters
LIWC	:	Linguistic Inquiry and Word Count
LPU	:	Learning from Positive and Unlabeled Examples
MCS	:	Maximum Content Similarity
MHCC	:	Multi-typed Heterogeneous Collective Classification
MPIPUL	:	Mixing Population and Individual Property PU Learning
PCFG	:	Probabilistic Context-Free Grammar
POS	:	Part of Speech
PU Learning	:	Positive and Unlabeled Learning
SPUL	:	Similarity-based PU Learning
SVM	:	Support Vector Machine
TF-IDF	:	Term Frequency-Inverse Document Frequency

REFERENCES

[1] J. Peng, K.-K. R. Choo, and H. Ashman, "Astroturfing detection in social media: Using binary n-gram analysis for authorship attribution," in *Proc. 15th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, 2016, pp. 121–128, doi: 10.1109/TrustCom/BigDataSE/ISPA.2016.53.

[2] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating Web spam with trustrank," in *Proc. 13th Int. Conf. Very Large Data Bases (VLDB)*, vol. 30. 2004, pp. 576–587.

[3] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam Web pages through content analysis," in *Proc. 15th Int. Conf. World Wide Web (WWW)*, 2006, pp. 83–92, doi: 10.1145/1135777.1135794.

[4] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999, doi: 10.1109/72.788645.
[5] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge, U.K.: Cambridge Univ. Press, 2015.
[6] J. Peng, K.-K. R. Choo, and H. Ashman, "Bit-level *n*-gram based forensic authorship analysis on social media: Identifying individuals from linguistic profiles," *J. Netw. Comput. Appl.*, vol. 70, pp. 171–182, Jul. 2016, doi: 10.1016/j.jnca.2016.04.001.
[7] J. Peng, S. Detchon, K.-K. R. Choo, and H. Ashman, "Astroturfing detection in social media: A binary n-gram–based approach," *Concurrency Comput., Pract. Exper.*, doi: 10.1002/cpe.4013.
[8] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.
[9] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, vol. 2. 2012, pp. 171–175.
[10] V. W. Feng and G. Hirst, "Detecting deceptive opinions with profile compatibility," in *Proc. 6th Int. Joint Conf. Natural Lang. Process. (IJCNLP)*, 2013, pp. 338–346.
[11] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proc. Int. Conf. Web Search Data Mining*, 2008, pp. 219–230, doi: 10.1145/1341531.1341560.
[12] A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal, "Detecting group review spam," in *Proc. 20th Int. Conf. Companion World Wide Web*, 2011, pp. 93–94, doi: 10.1145/1963192.1963240.
[13] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Proc. 7th Int. AAAI Conf. Weblogs Social Media*, 2013, pp. 409–418.
[14] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol.*, vol. 1. 2011, pp. 309–319.
[15] L. Zhou, Y. Shi, and D. Zhang, "A statistical language modeling approach to online deception detection," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 8, pp. 1077–1081, Aug. 2008, doi: 10.1109/TKDE.2007.190624.
[16] G. Wu, D. Greene, B. Smyth, and P. Cunningham, "Distortion as a validation criterion in the identification of suspicious reviews," in *Proc. 1st Workshop Social Media Anal.*, 2010, pp. 10–13, doi: 10.1145/1964858.1964860.
[17] R. Y. K. Lau, S. Liao, R. C.-W. Kwok, K. Xu, Y. Xia, and Y. Li, "Text mining and probabilistic language modeling for online review spam detection," *ACM Trans. Manage. Inf. Syst.*, vol. 2, no. 4, Dec. 2011, Art. no. 25, doi: 10.1145/2070710.2070716.
[18] Y. Lin, T. Zhu, X. Wang, J. Zhang, and A. Zhou, "Towards online review spam detection," in *Proc. 23rd Int. Conf. World Wide Web (WWW Companion)*, 2014, pp. 341–342, doi: 10.1145/2567948.2577293.
[19] Y. Lin, T. Zhu, H. Wu, J. Zhang, X. Wang, and A. Zhou, "Towards online anti-opinion spam: Spotting fake reviews from the review sequence," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 261–264, doi: 10.1109/ASONAM.2014.6921594.
[20] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 939–948, doi: 10.1145/1871437.1871557.
[21] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 823–831, doi: 10.1145/2339530.2339662.
[22] C. L. Lai, K. Q. Xu, R. Y. K. Lau, Y. Li, and L. Jing, "Toward a language modeling approach for consumer review spam detection," in *Proc. IEEE 7th Int. Conf. e-Bus. Eng. (ICEBE)*, Nov. 2010, pp. 1–8, doi: 10.1109/ICEBE.2010.47.
[23] S. Banerjee and A. Y. K. Chua, "Applauses in hotel reviews: Genuine or deceptive?" in *Proc. Sci. Inf. Conf. (SAI)*, Aug. 2014, pp. 938–942, doi: 10.1109/SAI.2014.6918299.
[24] N. H. Long, P. H. T. Nghia, and N. M. Vuong, "Opinion spam recognition method for online reviews using ontological features," *Tp Chí Khoa Hc*, vol. 61, pp. 44–59, 2014.
[25] M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in *Proc. 21st Int. Conf. World Wide Web (WWW)*, 2012, pp. 201–210, doi: 10.1145/2187836.2187864.
[26] F. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJAI)*, 2011, pp. 2488–2493.
[27] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100, doi: 10.1145/279943.279962.
[28] W. Liu, Y. Li, D. Tao, and Y. Wang, "A general framework for co-training and its applications," *Neurocomputing*, vol. 167, pp. 112–121, Nov. 2015, doi: 10.1016/j.neucom.2015.04.087.
[29] W. Zhang, C. Bu, T. Yoshida, and S. Zhang, "CoSpa: A co-training approach for spam review identification with support vector machine," *Information*, vol. 7, no. 1, p. 12, 2016, doi: 10.3390/info7010012.
[30] J. Chengzhang and D.-K. Kang, "Detecting the spam review using tri-training," in *Proc. 17th Int. Conf. Adv. Commun. Technol. (ICACT)*, Jul. 2015, pp. 374–377, doi: 10.1109/ICACT.2015.7224822.
[31] I. Ahmed, R. Ali, D. Guan, Y.-K. Lee, S. Lee, and T. Chung, "Semi-supervised learning using frequent itemset and ensemble learning for SMS classification," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1065–1073, Feb. 2015, doi: 10.1016/j.eswa.2014.08.054.
[32] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *Proc. 19th Int. Conf. Mach. Learn. (ICML)*, vol. 2. 2002, pp. 387–394.
[33] B. Liu, Y. Dai, X. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," in *Proc. 3rd IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2003, pp. 179–186, doi: 10.1109/ICDM.2003.1250918.
[34] D. Zhang and W. S. Lee, "A simple probabilistic approach to learning from positive and unlabeled examples," in *Proc. 5th Annu. UK Workshop Comput. Intell. (UKCI)*, 2005, pp. 83–87.
[35] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2008, pp. 213–220, doi: 10.1145/1401890.1401920.
[36] X. Li, P. S. Yu, B. Liu, and S.-K. Ng, "Positive unlabeled learning for data stream classification," in *Proc. SIAM Int. Conf. Data Mining*, 2009, pp. 257–268, doi: 10.1137/1.9781611972795.23.
[37] Y. Xiao, B. Liu, J. Yin, L. Cao, C. Zhang, and Z. Hao, "Similarity-based approach for positive and unlabelled learning," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 22. 2011, pp. 1577–1582, doi: 10.5591/978-1-57735-516-8/IJCAI11-265.
[38] D. Hernández, R. Guzmán, M. Móntes-y-Gomez, and P. Rosso, "Using PU-learning to detect deceptive opinion spam," in *Proc. 4th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2013, pp. 38–45.
[39] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *J. Big Data*, vol. 2, no. 1, p. 23, Dec. 2015, doi: 10.1186/s40537-015-0029-9.
[40] H. Li, B. Liu, A. Mukherjee, and J. Shao, "Spotting fake reviews using positive-unlabeled learning," *Comput. Sistemas*, vol. 18, no. 3, pp. 467–475, 2014, doi: 10.13053/CyS-18-3-2035.
[41] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao, "Spotting fake reviews via collective positive-unlabeled learning," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 899–904, doi: 10.1109/ICDM.2014.47.
[42] *LPU: Learning From Positive and Unlabeled Examples*, accessed on Oct. 7, 2003. [Online]. Available: https://www.cs.uic.edu/ liub/LPU/LPU-download.html
[43] A. Heydari, M. ali Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3634–3642, May 2015, doi: 10.1016/j.eswa.2014.12.029.
[44] Y. Ren, D. Ji, and H. Zhang, "Positive unlabeled learning for deceptive reviews detection," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 488–498.
[45] D. H. Fusilier, M. Montes-y-Gómez, P. Rosso, and R. G. Cabrera, "Detecting positive and negative deceptive opinions using PU-learning," *Inf. Process. Manage.*, vol. 51, no. 4, pp. 433–443, Jul. 2015, doi: 10.1016/j.ipm.2014.11.001.
[46] J. Karimpour, A. A. Noroozi, and S. Alizadeh, "Web spam detection by learning from small labeled samples," *Int. J. Comput. Appl.*, vol. 50, no. 21, pp. 1–5, Jul. 2012, doi: 10.5120/7924-0993.
[47] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002.

[48] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. 12th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

[49] B. Zhang and W. Zuo, "Reliable negative extracting based on kNN for learning from positive and unlabeled examples," *J. Comput.*, vol. 4, no. 1, pp. 94–101, 2009, doi: 10.4304/jcp.4.1.94-101.

[50] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2013, pp. 497–501.

[51] Y. Ren, D. Ji, L. Yin, and H. Zhang, "Finding deceptive opinion spam by correcting the mislabeled instances," *Chin. J. Electron.*, vol. 24, no. 1, pp. 52–57, 2015, doi: 10.1049/cje.2015.01.009.

JITENDRA KUMAR ROUT received the master's degree from the National Institute of Technology Rourkela, India, in 2013, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. His research interests include privacy in social networks, cryptography, natural language processing, and multimedia data mining.

ANMOL DALMIA is currently pursuing the master's degree in the field of information security with the Department of Computer Science and Engineering, National Institute of Technology Rourkela, India. His research interests include online and social security, expert mining, and computational trust.

KIM-KWANG RAYMOND CHOO received the Ph.D. degree in information security from the Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship with the University of Texas at San Antonio. He is a fellow of the Australian Computer Society. He was named one of ten Emerging Leaders in the Innovation category of The Weekend Australian Magazine / Microsoft's Next 100 series in 2009, and was a recipient of various awards, including the ESORICS 2015 Best Research Paper Award, the Highly Commended Award from Australia New Zealand Policing Advisory Agency, the British Computer Society's Wilkes Award, the Fulbright Scholarship, and the 2008 Australia Day Achievement Medallion. He serves on the Editorial Board of the *Cluster Computing*, the *Digital Investigation*, the IEEE CLOUD COMPUTING, the *Future Generation Computer Systems*, the *Journal of Network and Computer Applications*, and the *PLoS ONE*. He also serves as the Special Issue Guest Editor of the *ACM Transactions on Embedded Computing Systems* (2017; DOI: 10.1145/3015662), the *ACM Transactions on Internet Technology* (2016; DOI: 10.1145/3013520), the *Digital Investigation* (2016; DOI: 10.1016/j.diin.2016.08.003), the *Future Generation Computer Systems* (2016; DOI: 10.1016/j.future.2016.04.017), the IEEE CLOUD (2015; DOI: 10.1109/MCC.2015.84), the IEEE NETWORK (2016; DOI: 10.1109/MNET.2016.7764272), the *Journal of Computer and System Sciences* (2017; DOI: 10.1016/j.jcss.2016.09.001), the *Multimedia Tools and Applications* (2017; DOI: 10.1007/s11042-016-4081-z), and the *Pervasive and Mobile Computing* (2016; DOI: 10.1016/j.pmcj.2016.10.003).

SAMBIT BAKSHI received the Ph.D. degree in computer science and engineering in 2015. He is currently with the Centre for Computer Vision and Pattern Recognition, National Institute of Technology Rourkela, India, where he also serves as an Assistant Professor with the Department of Computer Science and Engineering. He has over 30 publications in journals, reports, and conferences. He is a Technical Committee Member of the IEEE Computer Society Technical Committee on Pattern Analysis and Machine Intelligence. He received the prestigious Innovative Student Projects Award–2011 from Indian National Academy of Engineering for his master's thesis. He currently serves as an Associate Editor of the IEEE ACCESS, the *Plos One*, the *Innovations in Systems and Software Engineering* (A NASA Journal), and the *International Journal of Biometrics*.

SANJAY KUMAR JENA received the M.Tech. degree in computer science and engineering from IIT Kharagpur in 1982 and the Ph.D. degree from IIT Bombay in 1990. He currently serves as a Professor with the Department of Computer Science and Engineering, National Institute of Technology Rourkela, India, where he is also serving as the Head of Training and Placement Cell. He is a senior member of ACM. His research interests include data engineering, information security, parallel computing, and privacy preserving techniques.