Using Boosting Approaches to Detect Spam Reviews

Sifat Ahmed
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
sifat.austech@outlook.com

Faisal Muhammad Shah
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
faisal505@hotmail.com

***Abstract*— When it comes down to buying products from online shops, one of the key factor that influences a buyer are the reviews associated with a product. While buying people try to understand the quality and authenticity of the product by reading the previous user feedback. And sellers have started taking advantage of it. Putting fake and spam reviews to deceive the buyers is a common strategy mostly used by newcomers. But these reviews are important when it comes to deciding whether to buy a product or not. We propose a method to detect these fake reviews from Amazon Review Dataset. Rather than using traditional machine learning classifiers we have used boosting algorithms to improve the accuracy of the traditional approach. In this approach, a significant increase in accuracy has been achieved by boosting weak learners. Up to 93% accuracy has been achieved when tried to detect fake reviews where traditional machine learning algorithms achieve an accuracy of up to 89%.**

Keywords—fake review detection, boosting algorithm, spam detection, active learning.

I. INTRODUCTION

For every product sold on the internet and every buyer bought a product, around 60% of people post a review of that. As people are getting more connected to the internet, the style, medium of putting a review has changed. Nowadays people even make video reviews of a product to attract sellers and the industry is still growing where people promote products for compensation. But amongst all, there are people who write honest reviews to help others when it comes to making a decision about buying the product. Some sellers are taking this advantage by posting fake reviews to deceive the buyers.

Detecting these fake reviews are not a simple task when it comes down to handing over the responsibility to the computer. Some fake reviews are so skillfully written that even a human can't detect that let alone computers. And the freedom to write anything with no monitoring has made it more difficult to detect these deceptive reviews. These deceptive reviewers are mainly known as opinion spammers and their activities are known as opinion spamming [5]. In most of the cases, the new sellers are taking the chance of opinion spamming to grow the business quickly. Some websites are also paying review writers to write these fake reviews [6]. As a result, detecting these deceptive reviews are necessary to maintain the trust issue between the buyer and the seller.

There are many researchers who have proposed several techniques to detect review spam. But there are some difficulties which have made this task more complicated. One of them is the unavailability of the real-life enriched labeled dataset. Most of the datasets are based on the pseudo-fake reviews which are different from the real world data. It's a major drawback when it comes down to work with the real-life data depending on the pseudo-review dataset.

In this paper, we first create a labeled dataset from real life data with the help of active learning. This solves the drawback which was discussed previously and enables us to perform an experiment on the real-life data. Traditional machine learning classifiers have been used by many researchers. But it didn't have a significant impact when it comes down to accurately detecting the real-life fake reviews. So we have introduced boosting algorithms such as Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost) and the Gradient Boosting Machine (GBM) in fake review detection side by side with the traditional machine algorithms.

Our methodology is different from the previous work for the following aspects-

- **Real life custom labeled dataset:** The dataset used in this experiment were manually prepared and labeled. Because of the lack of gold standard real-life dataset we had to go through this phase. And drastically it has made a significant impact on the accuracy of our model. From a chunk of 100,000 reviews, collected from the authors of [3, 4]. Then from the chunk, fake reviews were collected based on the writers, writing pattern, the time difference between reviews, types of reviews for a product. A keyword-based search was also done to find out keywords in reviews such as 'Fake Review', 'Review exchange for product', 'Honest Review'. These key-word based search added more versatility to the dataset.
- **Supervised boosting approach:** Supervised boosting is a new approach which is not widely applied in spam detection as traditional machine learning algorithms. Amongst all other types of classifiers such as tree-based, boosting, regression, ensemble, and neural network, boosting is on the rise among researchers.

978-1-7281-3445-1/19/$31.00 ©2019 IEEE

II. RELATED WORKS

Product reviews or consumer opinions are used in business as a guide for new customers. In e-commerce, product reviews written by customers can be beneficial. Positivity from the previous customer can bring new customers into the business. And it's one of the most vital things to generate revenue. Thus sellers always prefer a positive vibe of their product so that more customers get attracted. Throughout the process, some reviews come as opinion spam.

The term fake review was stated by the authors of [7] for the first time. In their research, they categorized it into different portions. The authors of [1] have used active learning to create a hybrid dataset from yelp and ott review dataset. Their proposed model performed very well on it. On the other hand, the authors of [8] and [9] explored time series and distributional analysis. From the crowdsourced fake review data Ott, Choi [9] focused on the fictitious opinions. In their model, they have used an n-grams based classifiers to detect the review spam. On the other hand, the authors of [10] proposed a novel approach of topic modeling using Latent Dirichlet Allocation. But the fixed number of topics was the main drawback when applied to the real-world dataset. Fei, Mukherjee [11] tried a different approach to ranking. They explored the burstiness in the reviews to detect the opinion spammers. The authors of [12] tried a graph relationship between the reviews and proposed a framework named FraudEagle. On an extended work [13], they collected the meta-data of the reviews to detect the review spammers.

Ren and Ji in [14] took the analysis one step further with the introduction of a convolutional neural network (CNN) and recurrent neural network. Their study showed that neural network based models have better generalization ability than discrete models. In [15], the authors proposed a heterogeneous graph model that finds the correlation between the reviews and the reviewers. This approach doesn't use the texts taken from the reviews. Therefore it only identifies the review spammer. In [16] the authors proposed a time series approach to detect the review spam. But the model suffered from a high computational problem during the scoring phase of the reviews.

III. DATASET & ACTIVE LEARNING

As stated before, a dataset with pseudo-reviews is available for analysis. But the problem with the dataset is that the model trained on this dataset didn't perform well on real-life test data. To solve the problem we created a hybrid dataset with the help of active learning. The active learning algorithm can interactively query the user or the information for the new data points to obtain the desired outputs and to reduce the cost of manual labeling of the dataset. A learner chooses a smaller set of examples to learn the concept. The main difference here with the normal supervised learning is that the learner learns from very small data samples than the one required for the normal supervised learning. We manually labeled 200 reviews as truthful reviews and another 200 as fake reviews. Then we fed these data to the learner to label the remaining 1600 data. For the remaining 1600 data, the model labeled 875 data samples as truthful reviews and leftover 725 reviews as fake reviews. So, we ended up with a dataset of 2000 samples where 1075 data samples were labeled as truthful reviews and 925 data samples were labeled as fake. Then this dataset was split into train, test and validation set. The algorithm of active learning is given below. The algorithm is quite same with authors proposed in [1].

Algorithm: Active learning process

```
INPUT:

initialTrain = The initial labeled training set;
initialTest = The initial labeled test set;
procData = unlabeled data;

OUTPUT:
finalTrain = hybrid training set;
//finalTrain    will    consist    intialTrainData    &
newLabeledData

1.    Load procData
2.    for all instances in procData
3.      create a sparse vector using the tf-idf vectorizer
4.       mat[] = sparse vector;
         //store s. Vector in a matrix
5.      feed mat[] to the CLASSIFIER
6.      accuracy = CLASSIFIER.accuracy
7.      if accuracy > 85% then
8.          finalTrain = (finalTrain U procData)
            //assert new labeled data
9.      else
10.       unlabData = DecisionFunction(procData instance)
11.       Manually label unlabData
```

Fig 1: Active Learning algorithm

IV. BOOSTING

The authors of [17] first came up with the concept of converting the weak learners into strong learner with higher accuracy. The work was then followed by the authors of [18] and [19] who made the boosting popular. They utilized it as a functional approximation of the logistic regression. This method of solving a problem is known as gradient boosting. This approach was slightly modified by the authors of [20] who fitted a subsample of the training set of the model without replacing it. This approach was based on the minimization of the loss function of the gradient and stochastic gradient descent in regression.

In [1], the authors introduced a hybrid ensemble approach to detect Amazon fake reviews. In their approach they have used active learning to create a labeled dataset first and then supervised learning with traditional classifiers such as Naïve Bayes, SVM, Decision Tree, Maximum Entropy and majority voting for classification were used. With n-gram features, their proposed model achieved an accuracy of 88%. Our proposed model extends their work to improve the performance with boosting. The extreme gradient boosting (XGBoost) is an end to end tree boosting algorithm which was proposed by the authors of [2]. The authors proposed it

as a novel sparsity-aware algorithm for sparse data and weighted quantile like a sketch for approximate tree learning. But the derivation of XGBoost follows from the existing idea of gradient boosting, originating a second order method from [Freidman at el]. The main factor of XGBoost is the leaf wise tree growth where the gradient boosting uses level wise tree growth. Recently XGBoost has become more popular because of winning kaggle competitions and outperforming classifiers and neural network models in some cases. 17 solutions out of 29 have used XGBoost and won the competition. 8 of the models solely used XGBoost while remaining models were ensembles of XGBoost and Neural Networks. Only some ensemble classifiers have outperformed well-tuned XGBoost. But they are very small in number according to [2]. Thus, this study and exploration of the previous work and kaggle winning solutions give us an overview that XGBoost can be explored with traditional approaches.

Our proposed method focuses more on XGBoost rather than other boosting approaches and classifiers that are commonly used. Different feature selection techniques were used to compare the performance of traditional classifiers and XGBoost. XGBoost performed brilliantly when compared with the previous work. Even with the default values of the hyperparameters described in [2], it beats the accuracy of fake review detection in a big margin. And finally, we ended up with a better result than the previous works.

V. METHODOLOGY

This section elaborates the experimental procedures that have been followed. We divide the whole methodology into 5 steps.

1. Collecting the unlabeled dataset.
2. Manually labeling some data and labeling remaining with Active Learning.
3. Preprocessing the dataset.
4. Feature selection.
5. Traditional and boosting approach to detect the review spam and comparison of results.

In previous researches, authors had their own way of preparing the dataset as gold standard data for amazon fake review detection is yet troublesome to find. That is where active learning played its role. Labeling all the data manually takes a huge effort, time and cost. To minimize these, we manually labeled some data with the help of experts and then using the active learning, labeled remaining dataset. Though it can't be claimed that our dataset is the gold standard but it is moderate enough for our research. From the dataset labeled by active learning, we picked some of the data and checked manually how active learning performed. Throughout the process, we labeled 1600 data with active learning and manually 400 data were labeled before. So we had 2000 labeled data where 1075 were labeled as truthful reviews and 925 reviews were detected as fake reviews. The dataset used here was collected from Amazon raw review data used by the authors of [3, 4]. It contains 142.8 million reviews and metadata spanning from 1996 – July 2016. The details of the dataset can be found at – http://jmcauley.ucsd.edu/data/amazon

After creating the dataset, in the second phase, we preprocess the dataset. The data we collected, needed a lot of preprocessing. The preprocessing part has not been discussed here as for the textual data standard preprocessing methods were followed such as removing stop words, stemming, lemmatizing etc. After preprocessing the dataset, it was split into 70%-10%-20% for training, validation, and testing. A small portion of the training set was used for validation.

In the feature selection phase, TF-IDF (Term frequency-Inverse document frequency) and Chi2 (Chi-Square) were used. TF-IDF increases the weight of uncommon words and decreases the weight of common word and results in creating a vector of features. On the other hand, Chi2 is a statistical approach that tries to find a relationship between every feature variable and the target variable. Features that are independent, gets discarded whereas dependent features get the most importance.

After the feature selection process traditional machine learning classifiers such as Decision Tree, Support Vector Machine, Stochastic Gradient Descent was trained and test data were used to check the accuracy. At the same time XGBoost and AdaBoost, GBM classifiers were trained and validated with training and validation sets and then accuracy was measured on the training set.

To understand the performance we took the accuracy, precision, recall and f1-score measurements. As said above, simulations were run based on different feature selection techniques, supervised approach with and without boosting, with default parameters and tuned hyperparameters. The results are discussed in the experiment section. The algorithm followed throughout the experiment is given below

Algorithm: Experimental Process

```
INPUT:

TrainData = The labeled training set (70%);
ValidationData = The validation dataset (10%)
TestData = Unlabeled dataset (20%)

OUTPUT:
Predictions = prediction from classifiers used.;
//ValidationData is used to validiate the classifier
predictions

1.    Load TrainData
2.    for all instances in TrainData
3.        create different sparse vector using the tf-idf
& chi2

4.        tf_idf_mat[] = sparse vector;
5.        Chi_2_mat[] = sparse vector;
6.        combined_mat[] = combine tf_idf_chi2_matrix
7.     for each feature matrix fed to the CLASSIFIER [DT,
SVM, SGD, XGB, ADB, GBM]
8.        train classifier
9.        accuracy, precision, recall = PREDICTION.metrics
10. RESULT COMPARISON
```

Fig 2: Experimental process Algorithm

VI. EXPERIMENT & RESULTS

In this part, we have discussed the details of the experimental procedure. As said above, using the active learning algorithm and manual labeling we created a dataset of 2000 reviews where 1075 were labeled as truthful reviews and the remaining 925 were labeled as fake reviews. We kept the size small to closely observe the performance. Before feeding this data into the classifiers, we need to select features from the data. In this case, tf-idf and chi2 were used individually. At first, we selected features with tf-idf and fed the data into traditional classifiers and boosting classifiers. In the second phase, we used chi2 as feature selection algorithm and followed the exact same procedure. At the third phase, we combined both to observe how all the classifiers perform. The boosting algorithms used here has several hyperparameters that can be tuned to achieve higher accuracy. Working with huge data can be time-consuming and costly when it comes to finding the best hyperparameter set. Choosing a smaller dataset of 2000 data has given us an advantage here. As for the traditional classifiers we have used

- Decision Tree (DT)
- Support Vector Machine (SVM)
- Stochastic Gradient Descent (SGD)

These traditional classifiers and boosting classifiers were simulated first before using Grid Search to find the best hyperparameter set. The results of the traditional classifiers with different feature selection technique are given below

Table 1: Performance of Traditional Classifiers on Tf-idf features

Classifiers	Feature Selection	DT	SVM	SGD
Accuracy	Tf-idf	0.824	0.833	**0.845**
Precision		0.810	0.845	**0.868**
Recall		0.804	0.860	**0.842**

The table above shows the performance metric for traditional classifiers with tf-idf. Tf-idf is one of the most popular feature selection technique used in natural language processing problems to select features. It can be seen that SGD achieves accuracy up to 84%. After observing performance with tf-idf feature selection performance with traditional classifiers, we then try chi2 features.

Table 2: Performance of Traditional Classifiers on Chi2 features

Classifiers	Feature Selection	DT	SVM	SGD
Accuracy	Chi2	0.846	0.857	**0.882**
Precision		0.829	0.875	**0.895**
Recall		0.839	0.872	**0.888**

Table 2 shows the performance of classifiers on chi2 features. From here we can say that the accuracy, precision, recall have improved. SGD achieved the highest among all up to 88% accuracy. Chi2 features have performed better than tf-idf features when fed into the target classifiers. After observing the accuracy of chi2 features, we combine both tf-idf and chi2 as part of our experiment to observe whether the metrics improve.

Table 3: Performance of Traditional Classifiers on combined tf-idf and chi2 features

Classifiers	Feature Selection	DT	SVM	SGD
Accuracy	Tf-idf + Chi2	0.842	0.868	**0.872**
Precision		0.813	0.845	**0.885**
Recall		0.798	0.879	**0.877**

From table 3, it can be observed that the combination of tf-idf and chi2 features didn't perform well. The overall accuracy was higher when only chi2 features were used. Here now on we have discussed the results of boosting algorithms with their default parameters and best parameter set found with Grid Search. The simulation procedure is the same as the traditional classification approach keeping the training data and feature selection techniques same for comparison purposes.

Table 4: Performance of Boosting Classifiers on Tf-idf features

Classifiers	Feature Selection	XGBoost	AdaBoost	GBM
Accuracy	Tf-idf	**0.932**	0.919	0.922
Precision		**0.947**	0.901	0.917
Recall		**0.911**	0.884	0.898

Boosting classifiers achieved a notable amount of higher accuracy than the traditional machine learning classifiers. These boosting classifiers ran on their default parameter values. XGBoost achieved up to 93% which is slightly higher than others with tf-idf features. In the table below the classifiers have the same configuration of parameters. Only the features were changed to chi2 to observe the performance.

Table 5: Performance of Boosting Classifiers on chi2 features

Classifiers	Feature Selection	XGBoost	AdaBoost	GBM
Accuracy	Chi2	**0.958**	0.942	0.952
Precision		**0.951**	0.911	0.939

Recall		**0.898**	0.887	0.906

As the traditional classifiers had slight improvement of accuracy in chi2 features from tf-idf features, boosting classifiers have also performed in the same way. Using chi2 features, evaluation metrics have slightly improved results. So combining tf-idf and chi2 features to observe the change in results.

Table 6: Performance of Boosting Classifiers on combined tf-idf and chi2 features

Classifiers	Feature Selection	XGBoost	AdaBoost	GBM
Accuracy	Tf-idf + Chi2	**0.937**	0.902	0.918
Precision		**0.932**	0.895	0.927
Recall		**0.909**	0.889	0.894

The combination of features causes the accuracy to be slightly reduced than only chi2 features. The accuracy achieved here is 93%. It can be said from the observations that all the classifiers performed well with chi2 features. The below figure demonstrates the overall highest accuracy achieved in the different experimental setup. It can be seen that XGB achieves the highest accuracy with Chi2 as a feature selection technique.

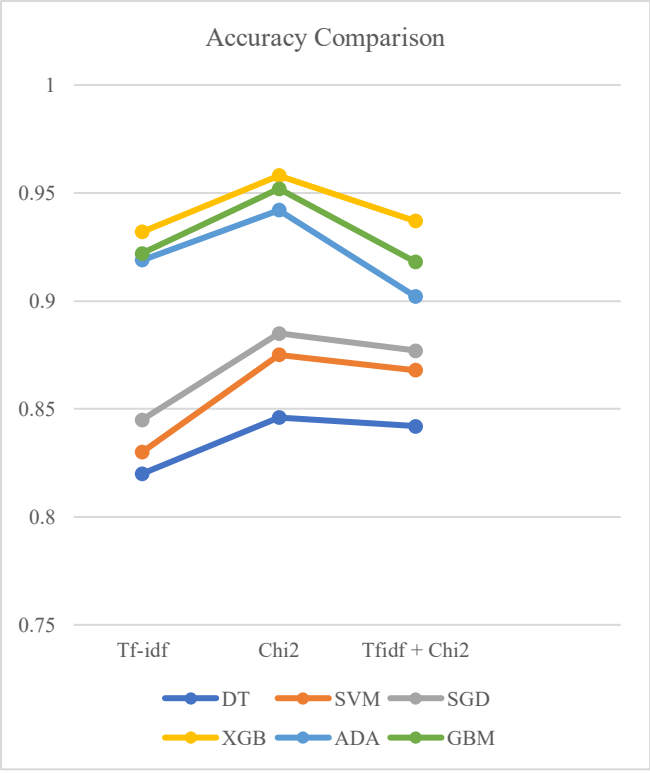

Fig 3: Accuracy Comparison

VII. CONCLUSION

As discussed earlier, we mentioned a lack of gold standard public dataset for analysis for which we had to introduce active learning. Manually labeling the data may increase the performance of the model. Using Extreme Gradient Boosting on fake review detection is still new and on the rise. There are many branches available to explore. As said, in our proposed method and experiments we didn't do deep hyperparameter tuning. Hyperparameter tuning is costly and time-consuming. Finding the best set of parameters can be very tricky. But with default parameters, it can be seen that XGBoost performed better achieving accuracy up to 95%. Finally, we hope to improve the dataset more and look forward to taking other aspects of the boosting algorithms to observe the change in results.

REFERENCES

[1] M.N. Istiaq Ahsan, Abdullah All Kafi, and Tamzid Nahian. Faisal Muhammad Shah, "An Ensemble approach to detect Review Spam using hybrid Machine Learning Technique." 2016 19th International Conference on Computer and Information Technology (ICCIT).

[2] Chen, Tianqi & Guestrin, Carlos. (2016). XGBoost: A Scalable Tree Boosting System. 785-794. 10.1145/2939672.2939785.

[3] Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering R. He, J. McAuley WWW, 2016

[4] Image-based recommendations on styles and substitutes. J. McAuley, C. Targett, J. Shi, A. van den Hengel SIGIR, 2015

[5] Algur, S., Hiremath, E., Patil, A. and Shivashankar, S., "Spam Detection of Customer Reviews from Web Pages." In Proceedings of the 2nd International Conference on IT and Business Intelligence.2010.

[6] Streitfeld, David. "Buy reviews on Yelp, get black mark." New York Times. Available:http://www.nytimes.com/2012/10/18/technology/yelp-tries-to-halt-deceptive-reviews. html. (2012)

[7] Jindal, Nitin, and Bing Liu. "Opinion spam and analysis." In Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 219-230. ACM, 2008.

[8] Xie, Sihong, et al. "Review spam detection via temporal pattern discovery."Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

[9] Feng, S., Xing, L., Gogar, A., and Choi, Y. "Distributional Footprints of Deceptive Product Reviews". ICWSM. 2012

[10] Kyungyup Daniel Lee, Kyungah Han and Sung-Hyon Myaeng. Capturing Word Choice Patterns with LDA for Fake Review Detection in Sentiment Analysis. WIMS 2016. Available at: https://dl.acm.org/citation.cfm?id=2912868

[11] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews," UIC-CS-03-2013. Tech. Rep., 2013.

[12] Jindal N, Liu B (2007) Review spam detection. In: Proceedings of the 16th international conference on World Wide Web (pp. 1189–1190). ACM, Lyon, France.

[13] Pennebaker, J.W. et al., The Development and Psychometric Properties of LIWC2007 The University of Texas at Austin. ,

pp.1–22.

[14] Ren Y, Ji D. Neural networks for deceptive opinion spam detection: An empirical study. Information Sciences. 2017;385–386:213–24.

[15] Wang G, Xie S, Liu B, Yu PS. Review Graph Based Online Store Review Spammer Detection. Proceedings of the 2011 IEEE 11th International Conference on Data Mining; Vancouver, Canada. 2118325: IEEE Computer Society; 2011. p. 1242–7.

[16] Heydari A, Tavakoli M, Salim N. Detection of fake opinions using time series. Expert Systems with Applications. 2016;58:83–92.

[17] Schapire RE. The Strength of Weak Learnability. Maching Learning. 1990;5(2):197–227.

[18] Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics. 2000;28(2):337–407.

[19] Friedman JH. Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics. 2001;29(5):1189–232.

[20] Friedman JH. Stochastic gradient boosting. Computational Statistics & Data Analysis. 2002;38(4):367–78.