Spam Review Detection Using K-Means Artificial Bee Colony

Prateek Saini, Sakshi Shringi, Nirmala Sharma, and Harish Sharma

Abstract The current businesses which use Internet for marketing depend on online reviews, and these online reviews can direct a customer toward or away from a product and service. This effect of online reviews is the reason that business uses spam reviews to either benefit their business or hinder their rival's business. In this paper, a novel solution k-means artificial bee colony for feature selection and optimized clusters using artificial bee colony to detect spam reviews is presented. We report the testing of our novel method on three different datasets. The findings of our testing are encouraging and show a respectable performance on all three datasets.

Keywords Artificial bee colony algorithm · Metaheuristic method · Spam detection · Spam reviews · K-Means · Machine learning

1 Introduction

In the last two decades, the Internet had changed the way we communicate, interact with our peers, and do business in a good way [1]. All this is possible because of the reachability of the Internet and its ease of use. The one major boom we see in the last decade is in e-commerce or in other words we can say including the power of Internet to increase the pool of available customers. This is done by providing products and services online which directly impacts the pool of customers meaning now the limit of accessing any service or product is limited only by the reachability of the internet. In all this, new currency for these online businesses emerges which is called product review or simply review.

One thing we can notice in today's online business that reviews play a major role whether a service or product will be able to make its place in the market or not [2]. Any customer who wants to buy a product or use a service through the internet will first go through the reviews of the respective product and service. At this point, if reviews do not give him enough confidence in the authenticity and reliability of the

P. Saini (✉) · S. Shringi · N. Sharma · H. Sharma
Rajasthan Technical University, Kota, India

H. Sharma et al. (eds.), *Communication and Intelligent Systems*, Lecture Notes in Networks and Systems 204, https://doi.org/10.1007/978-981-16-1089-9_57

product, it is highly likely that he/she will not buy it and look for something else or somewhere else.

A scenario-like mentioned above had encouraged the online businesses to understand the experience of their customer and improve their product or services to better serve them in the future. To do so a new technology at the time help them achieve such a goal, namely opinion mining [3]. Opinion mining allows businesses to analyze their customers by finding out their attitude toward their product or service. To do so, the opinion mining techniques use the reviews given by customers and try to find about their attitude based on those reviews but there is an underlying assumption that is made by these techniques which is that all reviews are trustworthy and authentic. But sadly that is not the case because most of the time each product page on the web is affected by 'spam reviews' which do affect these opinion mining techniques in negative way and defeat the purpose of using them in the first place.

Jindal and Liu [4] had categorized the spam reviews into three categories which have varying degrees of effect on the customer's decision to buy a product or not. 'Positive spam reviews' and 'negative spam reviews' are the most effective method to change a buyer's decision on false information. As the name suggests, 'positive spam review' gives a positive review to a non-deserving product and 'negative spam review' gives a negative review to a non-deserving product.

Consider a situation in which a restaurant failed to provide a nice dining experience to one of its customers and as a result of that the customer gives a bad review about the restaurant's bad hospitality on a site like Zomato or Yelp. As this can be bad for business, the restaurant hired someone to give positives reviews related to their dinning and mitigate the effect of that one honest review. These kinds of tactics cannot only be used to save your own skin but can also be used to destroy the business of your rivals. Spam reviews are designed to change the opinion of someone regarding something which is catastrophic.

In 2017, Rajamohana et al. [5] proposed a model using adaptive binary flower pollination algorithm for feature selection using naive Bayes classifier's accuracy as the objective function and k-nearest neighbors as the classifier using selected features. In 2019, Pandey and Rajpoot [6] proposed a model using spiral cuckoo search to optimize k-means algorithm using sum squared error as the objective function. The available literature is a source of motivation to carry out the future work in the field.

The rest of the paper is structured as follows: the artificial bee colony algorithm is reviewed in Sect. 2. In Sect. 3, k-means clustering is reviewed. Proposed feature selection and cluster head optimization using artificial bee colony are mentioned in Sect. 4. Experimental results are discussed in Sect. 5. Finally, conclusion is given in Sect. 6.

2 Artificial Bee Colony Algorithm

Artificial bee colony algorithm is a swarm-based algorithm which mimics the intelligent foraging behavior of honeybees. They forage honey by coordinating with each by

exchanging the location of the food source, i.e., honey. To exchange the information, they dance in a particular area of hive called the dancing area and the dance itself is called waggle dance [7]. This exchange of the information about the location of food source allows them to work collectively and efficiently. Their gathering, exchanging of information, and collective working are called collective intelligence, and it is the reason why we mimics their behavior to solve complex problems. Recently, ABC algorithm modified for various application and successfully applied to get rid of complex problem [8–10].

To understand the implementation of foraging behavior of honeybee, the whole process can be divided into four phases as discuss below.

2.1 Initialization Phase

This is the first phase of the algorithm and will be implemented only once. In this phase, position of search agents is randomly initialized within the search space according to Eq. (1).

Once the position has been initialized, then the fitness of each search agent is calculated.

$$x_i^j = x_{\min}^j + \text{rand}(0, 1)(x_{\max}^j - x_{\min}^j), \forall_j = 1, 2, \ldots, D \tag{1}$$

2.2 Employed Bee Phase

This is the second phase of the algorithm, and it will be implemented for each iteration of the algorithm. In this phase, each search agent changes their current position and evaluates the fitness of the new position. If the new position's quality is better than the current position, then it keeps the new one otherwise the old one. The new position is selected using Eq. (2)

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

2.3 Onlooker Bee Phase

This is the third phase of the algorithm, and it will also be implemented for every iteration. In this phase, a search agent is selected based on probability, on its fitness according to Eq. (3).

$$p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{N}} \tag{3}$$

Once a search agent has been selected, then the onlooker bee will change the current position and evaluate its fitness. If the new position is better than the current position, it will keep the new position otherwise the old one.

2.4 Scout Bee Phase

This is the fourth phase of the algorithm, and it will be implemented only when a search agent's position has not be changed for predetermined number of iterations. If any search agent enters this phase, it is now called a scout bee and it has to now find a new position. To do so, it is randomly initialized within the search space using Eq. (4).

$$x_i^j = x_{\min}^j + \text{rand}[0, 1](x_{\max}^j - x_{\min}^j), \forall_j = 1, 2, \ldots, D \tag{4}$$

3 K-Means Clustering

K-means clustering algorithm is designed to group similar things/samples/objects together or in other ways group them separately if they are dissimilar to each other in k distinct clusters. Their similarity and dissimilarity are evaluated based on what they are representing. If it is just numbers, then it can be their values or if its a complex object like word and image, it can be their attributes.

From implementation point of view, k-means clustering can be represented in simple four steps implemented in sequence to achieve the goal.

1. Select k points either randomly or form the samples such that they are not too close to each other and within the boundaries of sample space. These k points will represent the centroid of k clusters.
2. Now assign each sample to a cluster centroid which is closest to it.
3. For each cluster, calculate its new centroid by taking the mean of all the samples within the cluster.
4. Repeat steps (2) and (3) till there is no change in the position of cluster centroid.

4 Proposed Method

This paper introduced a clustering method optimized with ABC to detect spam reviews. The proposed method is divided into following phases.

1. Preprocessing
2. Feature Extraction
3. Proposed feature selection using k-means ABC
4. Proposed Artificial Bee Colony Optimizer with k-Means
5. Testing (Fig. 1).

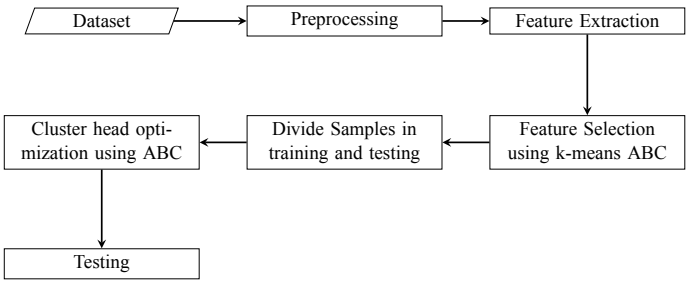

Fig. 1 Experiment process flowchart

4.1 Preprocessing

The online reviews most of the time contains noise or words that do not added any meaning to our model so instead of wasting resources in evaluating such entities we completely remove them out of the equation. Following are the operations performed in preprocessing

1. Convert all words into lowercase.
2. Remove stop words like is, or, as etc.
3. Remove symbols like pound, asterisk, etc.
4. Remove punctuation.
5. Replace continuous white spaces with one white space.

4.2 Feature Extraction

Feature extraction is done using linguistic inquiry and word count.

4.3 Feature Selection

Feature selection is used to remove redundant, noisy, and less significant features so that the learning model can give higher accuracy in a reasonable time. Higher-dimensional data generally tends to increase the training time without giving a significant increase in accuracy (sometimes making it even worse). For these two reasons, feature selection is desirable in most of the machine learning models.

K-means with ABC is used to find out the optimal feature set according to the Algorithm 1. In the proposed method, following steps are taken

1. Every search agent is initialized with random features, and their fitness values are calculated using Algorithm 2.

2. Algorithm 2 in turn uses Algorithm 3 for labeling the cluster heads and Algorithm 4 for calculating the accuracy.
3. In employed bee phase, each search agent's feature subset is changed by replacing one of the current feature with the new one. New fitness is calculated and better one among the current and new is kept.
4. In onlooker bee phase, search agents with higher probability of having a good solution are given the chance to update their feature set.
5. In scout bee phase, search agents whose feature set is not updated for a predefined number of times will be reinitialized randomly.
6. After maximum iteration has been done return the optimal set of features.

Algorithm 1: Feature selection

Input: number of search agents, dimension of search agents, data
Output: optimal features
`/* Initialization phase */`
while *every search agent didn't get a chance* **do**
  Initialize the search agents with random features
  Calculate the fitness of each search agent
end
while *maximum iteration not reached* **do**
  `/* employed bee phase */`
  while *every search agent didn't get a chance* **do**
    Select a search agent
    Replace one of its feature with a new one
    Calculate the new fitness
    Apply greedy selection on fitness
  end
  `/* onlooker bee phase */`
  Calculate probability of every search agent
  while *every search agent is not checked* **do**
    Select a search agent
    if *current search agent have good probability* **then**
      give it a chance to update its position
      apply greedy selection on the fitness
    end
  end
  `/* scout bee phase */`
  while *every search agent is not checked* **do**
    Select a search agent
    if *current search agent position not changed for a predefined number of times* **then**
      Randomly reinitialize the search agent
      Calculate fitness
    end
  end
end
Return the optimal feature set

Algorithm 2: Feature selection fitness function

Input: bee position, samples, samples label
Output: accuracy
Extract features from samples based on bee position
Apply kmeans on selected features
Label the cluster heads
Calculate accuracy
Return accuracy

Algorithm 3: Cluster labelling

Input: label of samples(label), to which cluster sample is assigned(belongs)
Output: cluster labels
while *all labels are not checked* **do**
  if *label is positive and belongs to cluster1* **then**
    cluster1.spam++
  else if *label is positive and belongs to cluster2* **then**
    cluster2.spam++
  else if *label is negative and belongs to cluster1* **then**
    cluster1.ham++
  else if *label is negative and belongs to cluster2* **then**
    cluster2.ham++
end
if *cluster1.spam* > *cluster2.spam* **then**
  label cluster1 as spam and cluster2 as ham
end
else
  label cluster1 as ham and cluster2 as spam
end
Return the label of cluster1 and cluster2

4.4 Divide Samples into Training and Testing

The original samples are divided into training samples and testing samples in 7:3 ratio.

4.5 Proposed Artificial Bee Colony Optimizer with K-Means

The proposed method uses ABC to find optimal cluster heads to classify reviews into spam and ham reviews. For finding optimal cluster heads, Algorithm 5 is used in which each search agent represents the position of two cluster heads spam and ham. Followings are the steps for Algorithm 5.

Algorithm 4: Accuracy Calculation Method

Input: label of samples (lable), label of cluster, to which cluster a sample bleong to(belongs)
Output: Accuracy

```
Initalize: TP, TN, FP, FN = 0, 0, 0, 0
while all samples are not checked do
    if label is positive and belongs to cluster1 and cluster1.label is spam then
        then TP++
    else if label is positive and belongs to cluster1 and cluster1.label is ham then
        then FN++
    else if label is positive and belongs to cluster2 and cluster1.label is spam then
        then TP++
    else if label is positive and belongs to cluster2 and cluster1.label is ham then
        then FN++
    else if label is negative and belongs to cluster1 and cluster1.label is spam then
        then FP++
    else if label is negative and belongs to cluster1 and cluster1.label is ham then
        then TN++
    else if label is negative and belongs to cluster2 and cluster2.label is spam then
        then FP++
    else if label is negative and belongs to cluster2 and cluster2.label is ham then
        then TN++
end
calculate accuracy
return accuracy
```

1. Initialize each search agent randomly within the search space and find the fitness using Algorithm 6.
2. Algorithm 6 will use Algorithm 3 for labeling the clusters and Algorithm 4 for calculating accuracy.
3. In employed bee phase, each search agent is given a chance to update its position using Eq. (2). New fitness value will be compared with the current fitness value, and the best will be kept.
4. In onlooker bee phase, search agents with higher probability of providing a good solution are given the chance to update their position. Probability for each search agent is calculated using Eq. (3).
5. In scout bee phase, search agents whose position is not updated for a predefined number of times will be reinitialized within the search space.
6. Return the optimal cluster position after maximum iterations are done.

4.6 Testing

For testing the efficiency of cluster heads provided by Algorithm 5, Algorithm 7 is used.

Algorithm 5: Cluster head optimization using ABC

Input: Number of clusters, data
Output: Optimum cluster heads
`/* Initialization phase */`
while *every search agent didn't get a chance* **do**
  Initialize the search agents with random values according to equation (1)
  Calculate the fitness of each search agent
end
while *maximum iteration not reached* **do**
  `/* employed bee phase */`
  while *every search agent didn't get a chance* **do**
    Select a search agent
    Replace one of its dimension with a new one according to equation (2)
    Calculate the new fitness
    Apply greedy selection on fitness
  end
  `/* onlooker bee phase */`
  Calculate probability of every search agent according to equation (3)
  while *every search agent is not checked* **do**
    Select a search agent
    if *current search agent have good probability* **then**
      give it a chance to update its solution
      apply greedy selection on the fitness
    end
  end
  `/* scout bee phase */`
  while *every search agent is not checked* **do**
    Select a search agent
    if *current search agent position not changed for a predefined number of times* **then**
      Randomly reinitialize the search agent
      Calculate fitness
    end
  end
end
Return the optimal cluster heads

Algorithm 6: Fitness function for cluster head optimization

Input: search agent position(position), samples, samples label
Output: accuracy
Extract cluster heads from position
Assign samples to cluster heads
Label cluster heads
Calculate accuracy
return accuracy

Algorithm 7: Testing

Input: search agent position(position), testing samples(samples), testing samples label
Output: accuracy
Extract cluster heads from position
Assign samples to cluster heads
Calculate accuracy
return accuracy

5 Experimental Results

The proposed method is tested on three datasets, namely Synthetic spam [11], Yelp [12, 13] and Movie [14], presented in Table 2. Synthetic spam dataset is taken from Database and Information system Laboratory, University of Illinois, and labeled using synthetic review spamming method. Yelp dataset is taken as a subset from restaurant and hotel data, and movie reviews are subset of IMDB dataset . All experiments are done on Python-3.6 on Intel core i5 processor with 6 GB of RAM (Table 1).

For calculating the effectiveness of proposed method, number of true positive, true negative, false positive, and false negative prediction are observed to calculate accuracy, precision, and recall.

- True positive represents spam review predicted correctly.
- True negative represents ham review predicted correctly.
- False positive represents ham reviews predicted incorrectly.
- False negative represents spam review predicted incorrectly.

These four parameters together represent confusion matrix and based on this confusion matrix, precision, recall, and accuracy are computed using Eqs. (5)–(7), respectively.

The proposed model is implemented a total of ten times on each dataset, and average values are considered to evaluate the overall performance of the model. Tables 3, 4, and 5 show the result of synthetic spam review dataset, movie review dataset, and Yelp dataset, respectively. Figure 2 shows the performance of proposed model on all three datasets with different size of feature set (Fig. 3; Tables 6 and 7).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{7}$$

Table 1 Parameters for k-Means ABC

S. No.	Parameters	Value
1	Population	30
2	Maximum iteration	500
3	Dimension	Two times number of features
4	Scout bee threshold	$\frac{\text{populationsize} * \text{Dimension}}{2}$

Table 2 Datasets and their composition

S. No.	Dataset	Total reviews	Spam	Ham
1	Synthetic spam	478	163	315
2	Yelp	4952	3709	1243
3	Movie	8544	3998	4546

Table 3 Synthetic spam review results

Run	TP	TN	FP	FN	Accuracy(%)	Precision(%)	Recall(%)	Time (s)
1	30	86	9	18	81.1188	76.9230	62.5000	464.0110
2	27	78	17	21	73.4265	61.3636	56.2500	461.9867
3	29	85	10	19	79.7202	74.3589	60.4166	465.2193
4	37	70	25	11	74.8251	59.6774	77.0833	478.1129
5	37	69	26	11	74.1258	58.7301	77.0833	463.4829
6	27	84	11	21	77.6223	71.0526	56.2500	464.4838
7	37	67	28	11	72.7272	56.9230	77.0833	460.0673
8	25	85	10	23	76.9230	71.4285	52.0833	460.5071
9	34	69	26	14	72.0279	56.6666	70.8333	462.4235
10	37	69	26	11	74.1258	58.7301	77.0833	461.1607
Average	–	–	–	–	75.6643	64.5854	66.6666	464.1455

Table 4 Movie results

Run	TP	TN	FP	FN	Accuracy(%)	Precision(%)	Recall(%)	Time (s)
1	707	823	540	492	59.7189	56.6960	58.9658	2849.0599
2	683	843	520	516	59.5628	56.7747	56.9641	2880.6256
3	670	842	521	529	59.0163	56.2552	55.8798	2869.1039
4	757	807	556	442	61.0460	57.6542	63.1359	2844.3945
5	813	793	570	386	62.6854	58.7852	67.8065	2897.2389
6	664	899	464	535	61.0070	58.8652	55.3794	2833.30398
7	695	874	489	504	61.2412	58.6993	57.9649	2832.4360
8	761	823	540	438	61.8266	58.4934	63.4695	2813.9950
9	663	875	488	536	60.0312	57.6020	55.2960	2860.8529
10	657	867	496	542	59.4842	56.9817	54.7956	2866.9748
Average	–	–	–	–	60.5620	57.6807	58.9658	2854.7985

Table 5 Yelp results

Run	TP	TN	FP	FN	Accuracy(%)	Precision(%)	Recall(%)	Time (s)
1	1103	2	370	8	74.5111	74.8811	99.2799	1858.1328
2	1104	4	368	7	74.7134	75.0000	99.3699	1888.4344
3	1109	1	371	2	74.8482	74.9324	99.8199	1835.2916
4	1111	0	372	0	74.9157	74.9157	100.0000	1840.2322
5	1111	0	372	0	74.9157	74.9157	100.0000	1877.7144
6	1060	29	343	51	73.4322	75.5523	95.4095	1840.2296
7	1102	4	368	9	74.5785	74.9659	99.1899	1858.2096
8	1111	0	372	0	74.9157	74.9157	100.0000	1850.6239
9	1110	0	372	1	74.8482	74.8987	99.9099	1884.0125
10	1109	1	371	2	74.8482	74.9324	99.819	1881.1705
Average	–	–	–	–	74.6527	74.9910	99.2799	1861.4052

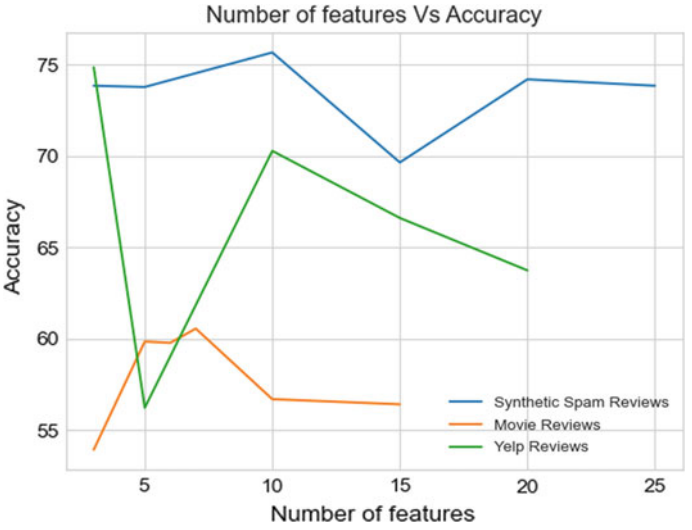

Fig. 2 Number of features versus accuracy graph

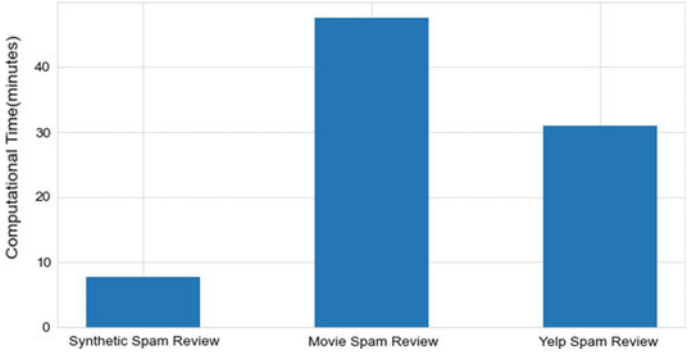

Fig. 3 Average computation time for one run

Table 6 Optimum number of features in each dataset

S. No.	Dataset	Optimum features
1	Synthetic spam review	11
2	Movie review	7
3	Yelp	3

Table 7 Average computation time for one run

S. No.	Dataset	Avg. computational time (min)
1	Synthetic spam review	7.7357
2	Movie review	47.5799
3	Yelp	31.0234

6 Conclusion

In this paper, we have introduced a novel approach by combining k-means and artificial bee colony algorithm for feature selection and cluster head optimization using ABC to detect spam reviews. The proposed method tested on three different datasets and gave us respectable results which shows the potential of the method. In our proposed model, we train on a snapshot of data which makes the model effective for current trend only. In future, the work can be extended to make the model continuously update its knowledge after a period of time. For feature selection, other optimization algorithms can be explored for more optimal set of features.

References

1. Van Deursen AJAM, Helsper EJ (2018) Collateral benefits of internet use: explaining the diverse outcomes of engaging with the internet. New Media Soc 20(7):2333–2351
2. Nieto J, Hernández-Maestro RM, Muñoz-Gallego PA (2014) Marketing decisions, customer reviews, and business performance: the use of the toprural website by Spanish rural lodging establishments. Tour Manage 45:115–123
3. Bakshi RK, Kaur N, Kaur R, Kaur G (2016) Opinion mining and sentiment analysis. In: 2016 3rd International conference on computing for sustainable global development (INDIACom). IEEE, pp 452–455
4. Jindal N, Liu B (2007) Analyzing and detecting review spam. In: Seventh IEEE International Conference on Data Mining (ICDM 2007). IEEE, pp 547–552
5. Rajamohana SP, Umamaheswari K, Abirami B (2017) Adaptive binary flower pollination algorithm for feature selection in review spam detection. In: International Conference on Innovations in Green Energy and Healthcare Technologies, pp 1–4
6. Pandey AC, Rajpoot DS (2019) Spam review detection using spiral cuckoo search clustering method Evol Intell 147–164
7. Karaboga Dervis, Akay Bahriye (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132

8. Sharma Sonal, Kumar Sandeep, Sharma Kavita (2019) Archimedean spiral based artificial bee colony algorithm. J Stat Manage Syst 22(7):1301–1313
9. Kumar S, Nayyar A, Kumari R (2019) Arrhenius artificial bee colony algorithm. In: International conference on innovative computing and communications Springer, pp 187–195
10. Nayyar A, Nguyen NG Kumari,R, Kumar S (2020) Robot path planning using modified artificial bee colony algorithm. In: Frontiers in intelligent computing: theory and applications. Springer, pp 25–36
11. Sun H, Morales A, Yan X (2013) Synthetic review spamming and defense. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1088–1096
12. Mukherjee A, Venkataraman V, Liu B, Glance, NS (2013) What yelp fake review filter might be doing? In: ICWSM, pp 409–418
13. Mukherjee A, Venkataraman V, Liu B, Glance N et al (2013) Fake review detection: classification and analysis of real and pseudo reviews. Technical report UIC-CS-2013–03. University of Illinois at Chicago
14. https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews