Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.046838

ARTICLE

An Online Fake Review Detection Approach Using Famous Machine Learning Algorithms

Asma Hassan Alshehri*

Department of Computer Sciences, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al Kharj, Saudi Arabia

*Corresponding Author: Asma Hassan Alshehri. Email: asm.alshehri@psau.edu.sa

Received: 16 October 2023 Accepted: 18 December 2023 Published: 27 February 2024

ABSTRACT

Online review platforms are becoming increasingly popular, encouraging dishonest merchants and service providers to deceive customers by creating fake reviews for their goods or services. Using Sybil accounts, bot farms, and real account purchases, immoral actors demonize rivals and advertise their goods. Most academic and industry efforts have been aimed at detecting fake/fraudulent product or service evaluations for years. The primary hurdle to identifying fraudulent reviews is the lack of a reliable means to distinguish fraudulent reviews from real ones. This paper adopts a semi-supervised machine learning method to detect fake reviews on any website, among other things. Online reviews are classified using a semi-supervised approach (PU-learning) since there is a shortage of labeled data, and they are dynamic. Then, classification is performed using the machine learning techniques Support Vector Machine (SVM) and Nave Bayes. The performance of the suggested system has been compared with standard works, and experimental findings are assessed using several assessment metrics.

KEYWORDS

Security; fake review; semi-supervised learning; ML algorithms; review detection

1 Introduction

The e-commerce business has exploded with the relevance of online evaluations, thanks to the fast expansion of internet access globally. In the current e-commerce sector, online reviews and ratings are being persuaded significantly. Customers sometimes depend significantly on reviews because they can't see the goods before buying them [1]. Online reviews are the main information source used by buyers to learn more about the items they want to purchase. Customer reactions to reviews may be both favorable and unfavorable, and they can also have a long-term good or bad influence on the company [2]. Consumers then analyze these reviews before making any decision related to purchasing [3]; thus, online reviews are marked as important information. Spam reviews highlight inappropriate information about the product or service offered by their competitors, and as a result, their products can be endorsed well [4]. Henceforth, detecting such reviews is crucial to help consumers make the right buying choices. Fake reviews are categorized into two classes [5]: unrealistic-these reviews sponsor or downgrade the product by associating negative or positive words to misguide the consumers. The second category is the review of brands, and reviewers recurrently use the brand name to promote a product [6].

Many techniques have been developed to detect these fake reviews, mainly relying on supervised Machine Learning (ML) methods and distinctive features. Implementing ML for fake review detection efficiently distinguishes between fake and genuine content [7]. Shan et al. [8] provided a technique based on internet customer reviews for detecting false reviews (OCR). Review inconsistency analyses, feature extraction, model construction, and sensitivity analysis are our four primary system components employed step-by-step to validate the classification and provide results. The authors of [9] detected bogus reviews using supervised machine learning techniques. The five classifiers employed are SVM, Naive Bayes, KNN, k-star, and decision trees. To identify fraudulent reviews on the dataset they had gathered, the authors in [10] employed Naive Bayes, Decision trees, SVM, Random forests, and Maximum Entropy classifiers. The authors employed both SVM and Naive base classifiers [11]. The yield dataset, which comprises 1600 evaluations gathered from 20 well-known Chicago hotels, was the focus of the authors' investigation.

The authors of [12] employed neural and discrete models using deep learning classifiers that included Average, CNN, RNN, GRNN, Average GRNN, and Bi-directional Average GRNN to identify misleading opinion spamming. It is very hard for anybody to manually read and effectively synthesize the vast number of reviews on various internet sites thus these activities and preventative measures require a competent fake review detector to be successful. The following list of challenges relates to several types of fraudulent review detection:

- It is difficult for humans and robots to review elements like ratings and brand name references.
- Rating behaviors are hard to discern when just one review is available for a certain item.

A technique for detecting false reviews that emphasizes PU learning and behavior density was developed. Most data sets are identified as unlabeled data U, while a minor portion is designated as P, the positive sample (Fake reviews). The letter RN appears on the trustworthy negative samples from U. The classifier is trained using the data sets P, RN, and assorted tester M (M = URN). The false behavior density of consumers and the false behavior density of apps are considered when calculating the behavior density. Finally, an SVM classifier is utilized to screen alleged false reviews further.

A collection of unlabeled data and a few positively labeled cases may be used using the semi-supervised method known as PU learning. Instead of just using instances with labels, this method uses many unlabeled data points. These tagged examples may be used to help produce labels for unidentified instances. Also, they can be used to train a classifier and assess a given review. In this paper, we use the semi-supervised learning approach (PU-learning) to enhance the classification of reviews and add three new features to the feature vector, including Point of Sale (POS) tagging LIWC, cosine similarity, and the n-grams frequency count, and behavioral features. The main contributions are as follows:

- We propose a novel fake/fraud review detection model in which SVM and NB algorithms capture the product-related review features and establish a classifier based on the product. A Yelp dataset including both negative and positive reviews is used to train a semi-supervised PU learning and machine learning algorithm to identify false reviews.
- A collection of unlabeled data and a few positively labeled cases uses the semi-supervised method known as PU learning. Instead of just using instances with labels, this method uses many unlabeled data points.
- We conducted several tests to evaluate the performance of our model. We reaffirm that our methodology is superior to cutting-edge techniques.

The remainder of the paper is organized as follows. The literature review is presented in Section 2, and the recommended strategy is described in Section 3. The dataset, the method, and the feature selection are all described in Section 3. The experimental findings are shown in Section 4, and Section 5 finishes the complete study that has been given.

2 Literature Review

The issue of fraudulent internet reviews is becoming more widely acknowledged. Positive and negative reviews elevate or denigrate specific goods to influence customers' choices and gain a competitive edge. Molla et al. [10] used a hybrid approach combining behavior-based features and content. The fake reviews are detected using machine learning classifiers, giving input of 133 unique features generated from a combination of content and behavior-based features. The data set is generated from Yelp.com. Feature extraction tactics focus on linguistic characteristics of content and reviewers' behavior. Together, these approaches provided a good result in the fake review class, and accuracy for both large datasets was attained by the CNN Type 1 and GB collective, which also did fine with the smaller datasets. However, this approach cannot identify fake reviews from diverse applications.

Vidanagama et al. [11] proposed a system detecting fake reviews by analyzing review and reviewer-centric features through a supervised classifier based on Random forests. The features used are (Textual features, Meta-Data features, burst features, rating features, and temporal features). After feature extraction, the Random Forest classifier uses a balanced test data set to avoid filtration of genuine reviews. The results shown are achieved by considering the Yelp Zip dataset, five cross-validations have been applied to omit overfitting as reviews from the same user can appear both in training and test sets. Ren et al. [12] examined the issue of false reviews and provide a broad learning goal for PU learning that focuses on particular unbalanced data. The authors proposed an estimate error constraint and conceptually demonstrated that, in terms of expectation, optimizing our learning goal is comparable to learning a classifier on the balanced, oversampled data with both P and N data available. Both the techniques discussed above are limited by the scope they consider to detect fake reviews.

He et al. [13] projected a system using four fake review data sets of hotels, restaurants, Yelp, and Amazon. Different sizes of data sets and several input word-embedding matrices of n-gram features of the review's text are created. In this approach, the hidden and varied qualities, misleading, false reviews have long been a challenge in fake review identification. According to Bhudi et al. [14], a fake review detection model based on sentiment intensity and PU learning (SIPUL) was proposed to address the abovementioned issues. SIPUL can continuously train the prediction model from the continually entering data stream. First, the sentiment intensity was introduced to categorize the reviews into distinct subsets (i.e., strong and weak sentiment sets) as soon as the streaming data arrives.

Fontanarava et al. [15] emphasized product-related review features for fraudulent review identification. The cosine similarity between each iteration review for the same product and review for other items is determined (based on the bag of words from the two reviews)—the model of bagging three classifiers (PWCC, TRIGRAMSSVM, BIGRAMSSVM). WCC is a CNN model incorporating product-related review elements to win these models. The CNN average pooling layer uses the product word composition (PWC) vectors as input to create a document model. The fake Review Detection Framework (FRDF) presented by Kauffmann et al. [16] recognizes and eradicates fake reviews by applying NLP knowledge. The review phrases were broken into tokens and marked with lexical, syntactic, and semantic data during pre-processing. Additionally, sentiment analysis aims to determine the degree of positivity or negativity of the remark concerning the product based on the words used.

A method used by FRDF is the reflection of phony reviews for two or more similar evaluations from separate reviewers or different goods.

Due to the serious and pervasive problem that fake reviews represent, it is still vital yet challenging to assist customers and businesses in telling the difference between genuine and fraudulent reviews. Heuristic methods, supervised machine learning, and manual efforts may all be used to detect bogus reviews. Certain techniques in the literature are only predicated on traits extracted from the review text. Even though detection investigations have progressed, several challenges remain. It is necessary to enhance categorization performance to keep up with text-generation algorithms. It is conceivable that examples with inaccurate labels exist in the databases, that their design is flawed, or that they are not publicly accessible.

3 Proposed Method

Some of the methods used in sentiment analysis include Natural Language Processing (NLP), Machine Learning, Text Mining, Information Theory and Coding, and the Semantic Approach. Subjective information in text analytics and NLP refers to natural language utterances expressing opinions, feelings, points of view, and attitudes toward a particular subject of interest. The method of mechanically analyzing these utterances to ascertain the sentiment expressed there is known as sentiment analysis, and it is intriguing from both a scientific and an industry perspective. This approach may process large volumes of data to track society's attitudes about public issues, events, or items. In the late 1990s, when the first useful datasets for subjective categorization were made available and the first paradigms that made it feasible to develop models that successfully addressed the issue were put out, SA began to gain popularity.

Sentiment analysis is a Natural Language Processing (NLP) technique utilized to detect the polarity of data, whether negative, positive, or neutral. It is mainly performed on textual data to assist corporations in examining brand name and product sentiment in consumer feedback. Consumers convey their views more openly these days; hence sentiment analysis is a vital tool for understanding the sentiment related to the product. Considering the era of e-commerce, analyzing product reviews is crucial. This paper uses semi-supervised and machine-learning models to detect fake reviews in the online product dataset. The sentiments associated with a review greatly help in identifying its authenticity. These techniques are also established based on a consistent criterion, thus easing the process of fake review detection by improving its accuracy. The proposed method consists of data collection, preprocessing, feature extraction, selection, and classification. Data preprocessing: Text data requires preparation before applying it to classification models. The basic feature extraction using NLP finds the similarities between portions of text and allows the computers to process human (natural) language. Three main steps are involved in data preprocessing: tokenization, normalization, and vectorization.

Three major components for fake review detection are ML algorithms, features, and the dataset. In our system, the tactics associated with these components are stated in Fig. 1, which shows the architecture of the proposed approach.

- Features: The features selected for this system are N-gram, sentiment score through LIWC, POS tagging, multiple behavioral features, and cosine similarity between product description and reviews.
- Dataset: Publicly available data sets from Yelp and Amazon.

- ML algorithms: We have opted for semi-supervised PU learning suitable for partially labeled data to identify fake reviews. We also used iterative SVM-based classification and Naive Bayes to classify reviews as fake or genuine.

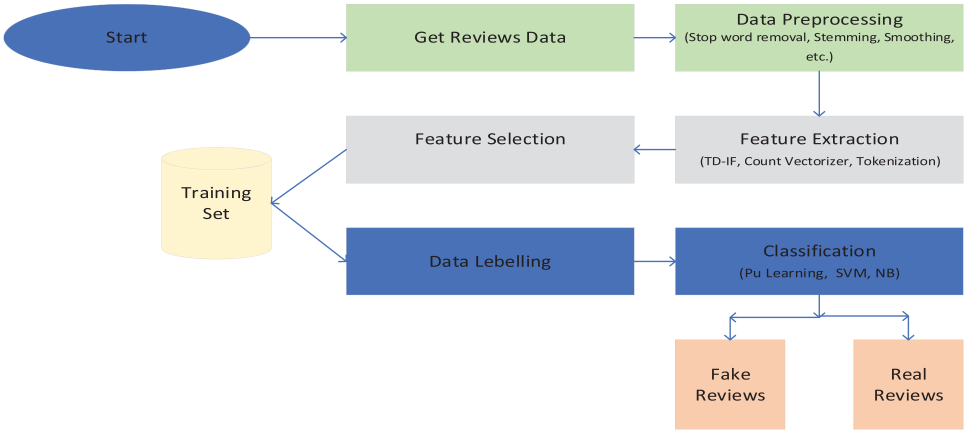


Figure 1: The architecture of the proposed system

3.1 Feature Selection

Fake reviews are divided into two categories: untruthful brand reviews and untruthful product reviews. Untruthful reviews are used to defame or promote a certain product with positive or negative words to misguide the consumers. The sentiment analysis score identifies these reviews by separating the words from the review text according to their polarity. In contrast, a review on a brand aims to promote a brand frequently, detected by checking the frequency of words captured in certain reviews [17]. A hybrid system for sentiment analysis has been opted for, which will first apply rule-based sentiment analysis, and later the automatic system will be incorporated into the process for machine learning-based classification. The data based on three types of features is given as input to the PU learning classifier for binary classification. The feature extraction is carried out by count vectorizer and TD-IF vectorization, which will generate feature vectors given as input to the model returning predicted labels (fake or genuine). The three basic methods [18,19] for detecting fake reviews are content-based, reviewer-based, and product-based approaches, as stated below.

3.1.1 Content-Based

It focuses on the text of the review. It covers the linguistic features, parts of speech (POS), linguistic inquiry and word count (LIWC), and text categorization by n-grams to build features for classification. Languages and customs are written down for linguistic characteristics. N-gram, POS, LIWC, and stylistic traits are among the linguistic and literary aspects [20]. The Unigram will divide the review content into a single word set; the bigram generates a set of two successive words, and the trigram gives a set of three consecutive words. Using this approach, we can discover the frequency of each word for reviews which will be added to our feature vector and later used for classification.

- Bigram: [a decent] [video and] [We have] [no objections] [regarding this] [one].
- Trigram: [a decent video] [and We have] [no criticisms] [regarding this one].

The POS tagger uses indications linked to review requests to analyze grammatical fraud to identify grammatical data in each paragraph word. The use of imaginary words, pronouns, and verbs is greater in fake reviews, while genuine ones are more explanatory and use more adjectives or nouns. Tagging (a type of classification) is the automatic assignment of the description of the tokens. We call the descriptors 'tag', which represents semantic information.

3.1.2 Linguistic Inquiry and Word Count (LIWC)

As a classification tool for fraudulent reviews, positive and negative emotion scores and punctuation mark scores are examples of LIWC characteristics [21]. Experimental outcomes using LIWC validate its capacity to sense meaning in an extensive variability of experimental situations, comprising a display of attentional emphasis, emotionality, social relations, thinking classes, and individual alterations [22]. The LIWC is used to create a sentiment score of the given review text; the classes used are positive and negative, as shown below in Fig. 2.

Review Text
LIWC
Positive
&
Negative sentiment
Score

Figure 2: LIWC sentiment score

We have assigned 1 to positive sentiments and 0 to negative ones. The extracted sentiment scores by LIWC are used to create feature vectors through vectorization. The generated feature vectors are then used in classification to label a review as fake or genuine. Moreover, we have added stylistic-based features that hinge on a word similarity measure (cosine similarity) and semantic similarity among products and reviews.

3.1.3 Behavior-Based

In fake reviews, the spammer mirrors their thoughts and emotions differently than a genuine reviewer. They write many fake reviews from varying social media accounts rather than choosing different time intervals. Spammers' average and maximum posting rates are higher as they aim to create an effective impression of their goal to defame a certain product or service. In connotation to this goal, they also give an unfair star rating to the product [23]. We have used a composed model to integrate multiple behavioral features for spammer identification. The age, profile, and URL through which the user has given a review are tracked to link the review to a certain user profile.

3.1.4 Product-Based

Fake reviews have more positive/negative sentiments than genuine ones [24], and the product is reflected in extreme wording to defame or promote it. The fake review detection model can use these product-related reviews [9]. They can greatly influence the prediction's performance, making it beneficial for the classification model. The product-related review features are generated by calculating the cosine similarity between the review and the product. The cosine similarity is calculated based on the bag-of-words of two reviews, the vectors for each sample are created through TF-IDF vectorization. In cosine similarity, the product-related reviews in amazon's dataset are treated as a vector. Fontanarava et al. [15] stated that the content similarities between two reviews about the same products are greater than those of different products. Different types of features are shown in Fig. 3. The formula used to calculate the cosine similarity between these two vectors is as follows:

- M = vector of reviews on the same product
- N = vector of reviews on different products
- N = dot product of the vectors ‘M’ and ‘N’
- ||M|| and ||N|| = length of the vectors ‘M’ and ‘N’.
- ||M|| ∗ ||N|| = cross product of the two vectors ‘M’ and ‘N’

Using machine learning models or training is not necessary using an approach based on rules for evaluating text. This approach results in guidelines that may categorize a text as positive, negative, or neutral. These guidelines are sometimes referred to as lexicons. Instead of rule-based systems, automated approaches use machine learning techniques rather than manually defined rules. A classification challenge is how sentiment analysis problems are often defined, where a classifier is given a text and returns a category, such as positive, negative, or neutral.

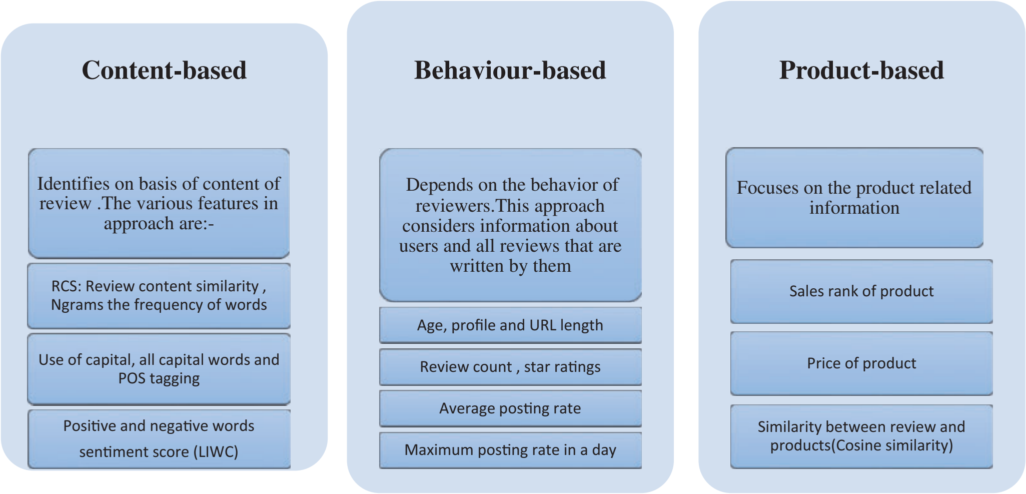

Content-based
Identifies on basis of content of review .The various features in approach are:-
RCS: Review content similarity , Ngrams the frequency of words
Use of capital, all capital words and POS tagging
Positive and negative words sentiment score (LIWC)
Behaviour-based
Depends on the behavior of reviewers.This approach considers information about users and all reviews that are written by them
Age, profile and URL length
Review count , star ratings
Average posting rate
Maximum posting rate in a day
Product-based
Focuses on the product related information
Sales rank of product
Price of product
Similarity between review and products(Cosine similarity)

Figure 3: Types of features

3.2 Pre-Processing of Data

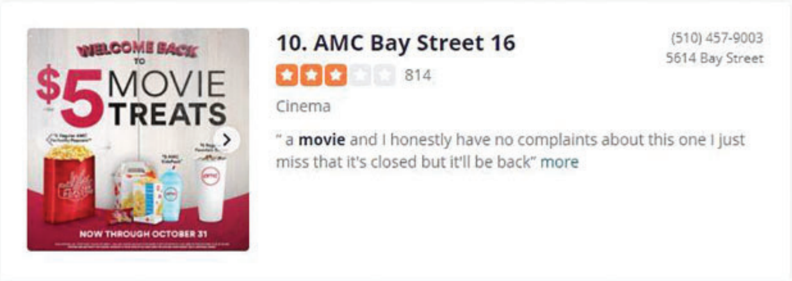
As part of the data mining and analysis process, data pre-processing converts raw data into a format that computers and machine learning can understand and comprehend. Data preparation includes cleaning, instance selection, normalization, transformation, feature extraction, and selection. During the data preparation process, the final training set is created.

• Data Cleaning/Cleansing: Real-world data is typically unreliable, noisy, and inconsistent. Data cleaning techniques try to detect outliers while finding missing data and removing noise.

• Data Transformation: Data is converted into mining forms appropriate to the circumstance.

• Data Reduction: Large datasets may need complex data processing and mining; doing research like this is expensive or impractical. Data reduction techniques are useful for obtaining qualitative information and evaluating a condensed version of a data collection while preserving the integrity of the original data.

• Tokenization: The longer sentences of review text are split into smaller pieces or tokens. It is also denoted as lexical analysis or text segmentation. It converts text into tokens before converting it into vectors. In Fig. 4, a review of a movie theatre on Yelp.com is shown, in this step, we will tokenize the text into words. The output achieved after tokenization is [‘a’, ‘movie’, ‘and’ ‘I’, ‘honestly’, ‘have’, ‘no’, ‘complaints’, ‘about’, ‘this’, ‘one’, ‘I’, ‘just’, ‘miss’, ‘that’, ‘it’s’, ‘closed’, ‘but’ ‘I’ll’, ‘be’, ‘back’]. Secondly, stopping words is removed by utilizing the NLTK (Natural language toolkit) library. Stop word removal is necessary as these words are irrelevant in the data context and no deeper meaning related to the sentence is obtained.



10. AMC Bay Street 16

(510) 457-9003
5614 Bay Street

★★★☆☆ 814

Cinema

" a **movie** and I honestly have no complaints about this one I just miss that it's closed but it'll be back" more

Figure 4: Review of a movie theater on Yelp.com

• Normalization: Generally, it refers to a set of related operations aimed at putting all text on the same level, such as converting all text to the same case (upper or lower), deleting punctuation, and converting numerals to their word counterparts. Normalization equalizes all words and allows processing to proceed consistently. To make the data ready for classification and before proceeding towards vectorization, the following steps were followed to normalize the data:

1. Casing the Characters: Conversion of all characters to lowercase for simultaneous recognition of words.
2. Negation Handling: The public reviews on Amazon and Yelp are full of apostrophes and connecting words; hence, they are converted to standard lexicons, resulting in a grammar-free context.
3. Removal: After tokenization, separate punctuation, special characters, and numerical tokens are removed to avoid their inclusion in sentiment.

• Count Vectorizer: It is a utility made available by the Python sci-kit learn package. Calculating each word’s frequency across the full text converts a given text into a vector. It will produce a matrix where each row is a text sample from the review content document and each column represents a unique term in the review. The cell’s value is just the number of times that term appears in the review text. This function returns an encoded vector containing the length of the whole vocabulary and an integer count of how many times each word appears in the input file.

The count vectorizer ignores single characters in a sparse matrix generated after tokenization and the stop words list created through the genism package is passed as an argument. The less significant stop words are also removed. Later min-df function is also applied to remove less frequent words appearing only once or twice, thus called noise. Fake reviews frequently use similar words in their content to give more heed to their point. Finally, we have used n-grams to generate the feature vector based on the frequency of words. The n-gram range is set to 3, specifying that it will use unigrams (single words), bigrams (a combination of two words), and trigrams (a combination of three consecutive words). Using this approach, we can discover the frequency of each word for reviews which will be added to our feature vector and later used for classification.

3.3 Dataset

Many e-commerce websites like Yelp and Amazon allow users to write reviews about certain products or services. The data is divided into three types: Review content, User behavior, and product-related information. Yelp was launched in 2005 and it is a source for running businesses. It has over one million business information and 135 plus million reviews from multiple reviewers. Yelp already uses a filtering algorithm to detect fake reviews, but these techniques are not public. We used the standard gold data set developed by Ott et al. to detect fake reviews. This data set comprises 1600 reviews on hotels in Chicago; 800 reviews are fake, while the remaining 800 are genuine. Fake and genuine reviews are differentiated by associating a binary tag with each category. Tag 0 is assigned to fake reviews and 1 denotes a real review. The data set of genuine reviews is divided into half, 400 reviews are assigned positive sentiment polarity.

In contrast, the remaining ones are assigned a negative sentiment polarity and a similar technique is used for the deceptive reviews class. The reviews are collected from various resources like Amazon Mechanical Turk (AMT) helped in generating fake reviews and remaining online reviews like Yelp, hotels.com, etc. The dataset is partitioned by random sampling to create a training and test set. Each member had an equal probability of being chosen in the random sample.

3.4 Machine Learning Algorithms

3.4.1 SVM

SVM is a technique for grouping data that addresses classification problems. The most current data mobility is separated into two groups, and the technique is gathered from the beginning for whatever is helpful. Each additional data point's class membership is estimated using an SVM. his classifier is a semi-equal automated technique. Using a flat hyperplane between two groups is far more challenging to control in this strategy. It is uncertain if this strategy will include a classification element to identify the hyperplane.

SVM is a binary classification algorithm that classifies samples into one or two classes (fake or genuine). It can also perform well with minimal data, the positive instances from the data set. SVM classifies samples by discovering the extreme margin hyperplane that can categorize the cluster of text's features amongst two classes (0 or 1), the maximum space between the hyperplane and the nearest sample from either class. The advantages of SVM are that it is effective in high dimensional space and uses a subset of training points, making it memory-efficient and adaptable.

3.4.2 Naive Bayes

Naive Bayes is a probabilistic multi-class classification algorithm relying on Bayes' theorem, and it assumes that input features are independent before predicting an output class. The training data is utilized to estimate the probability of features belonging to a specific class: LIWC, POS tagging, N-grams, and cosine similarity. To predict the class testing data by generating the probability value for features part of the current classes. The root of this machine learning model is the Bayes theorem. The Classifiers are models that assign a target class to troubling circumstances and are described as sets of extracted attributes, which are made using the Naive Bayes technique. The Bayes classifier is frequently utilized due to its superior performance compared to more advanced classifier methods. The following is how Naive Bayes works with the Bayes theorem's general principle:

In Naive Bayes, T is the hypothesis and S is the dataset.

- P(T|S), P(S|T) are the probability of hypothesis T given the dataset S and the probability of dataset S given the hypothesis T.
- P(T), P(S) are known as the probability of hypothesis T and the probability of dataset S.

The maximum posterior hypothesis T can be defined in Eq. (1).

$$\mathrm{MAP(T)} = \max((\mathrm{P(S|T)} * \mathrm{P(T)})/\mathrm{P(S)} \tag{1}$$

Identifying fake reviews requires data classification, but there are mostly inadequate labels. Physically labeling the data can lead to human favoritism or substantial errors. PU learning is implemented in a scenario where the data is labeled for a positive class and has unreliable labels for a negative class. It is a semi-supervised binary classification technique that mends even from undetermined cases in the data by learning from the positive cases. The information gathered from positive cases is then applied to relabeling the unknown cases. Two different sorts of PU learning versions are produced by using unlabeled data. Both techniques build the final classifier using uplifting examples. One of these variants makes very few samples from the unlabeled set [25,26], whereas the other one produces a classifier based on the complete unlabeled sample [27,28]. The training procedure consists of positive and unlabeled data and the assumption is made that an unlabeled sample is either a part of a negative or positive class. This algorithm attained recognition because it does not needfully supervise data and can also work for a few available labeled examples.

3.4.3 PU Learning

In binary classification, the classifier is trained to distinguish between two classes based on their attributes. The objective is to construct a binary classifier to categorize the test set T into two classes, positive and negative, where T can be U. This is given a set of examples of a specific class P (called the positive class) and a set of unlabeled examples U, which contains both class P and non-class P (called the negative class) instances. The positive set, P, and the mixed set, U, which is expected to include both positive and negative samples but not labeled as such, are the two sets of examples considered accessible for training in PU learning. In contrast, other types of semi-supervised learning presume the availability of both unlabeled samples and a labeled collection containing instances of both classes. Several methods are available to modify supervised classifiers for the PU learning environment, including variations of the EM algorithm.

Only the input data set includes the unlabeled sample U and the positive sample P (false reviews). Unlabeled data is used to identify the trustworthy negative sample (TS) for classifier learning. This study uses the PU algorithm to determine the trusted negative sample in order to choose the representative trusted negative cases more effectively. Every training sample is a tuple (a, b) where "a" defines vector attribute values and "b" is the class value. Therefore, the aim of PU learning is also similar to binary classification, it involves training a classifier that can eventually differentiate between fake and genuine review classes based on attributes. Only a few positive and negative samples are not yet utilized in the learning phase. We have used a data set (Gold standard data set) that previous researchers already labeled, thus helping us predict the values of S while selecting a tuple from the raw data set. The text has also already been processed for feature extraction and numeric values are generated from the attributes. The data set is merged to create a dictionary and map the numeric values to begin the classification process. In this paper, attributes selected for feature extraction and used for the experiment include the following:

- N-grams (bigram, trigram) frequency count
- POS tagging

- LIWC
- Cosine similarity
- Behavioral features

From Fig. 5, we can see that, while working on the ith review, the equivalent features are created in the following procedure:

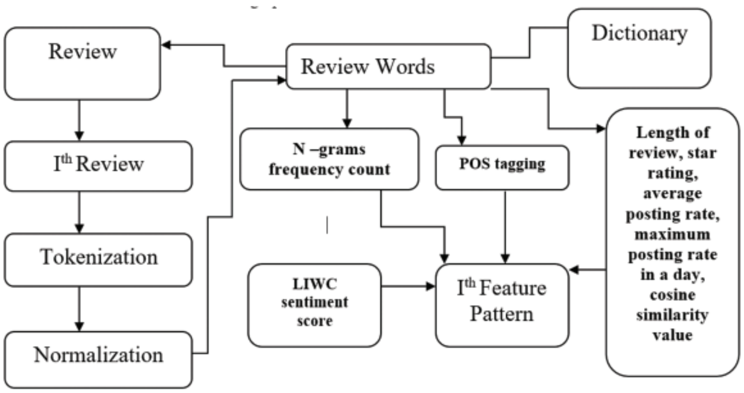


Figure 5: Equivalent feature extraction

1. To make a review eligible for vectorization, it goes through the tokenization process, and then normalization is performed. This outcome is the generation of nominee feature words.

2. Its frequency count is measured using n-grams and updated to the column in the vector conforming to the numeric map of the word.

3. The length of review, star rating, average posting rate, maximum posting rate in a day, cosine similarity value, and other behavior, product features are added to the feature vector.

4. The sentiment score generated through LIWC is also added to the feature vector. We have assigned 1 to positive sentiments and 0 to negative ones. The extracted sentiment scores by LIWC are used to create feature vectors through vectorization.

1) Average and Maximum Posting Rate: It displays the proportion of a reviewer's overall reviews to their number of active days (see Eq. (2)). A reviewer considers a day active if they have posted at least one review. Eq. (3) illustrates the Maximum Posting Rate, which is the maximum number of reviews that may be submitted in a single day.

$$\mathrm{APR}(a) = \mathrm{Nr}(a)/\mathrm{N}(\text{posting days}) \tag{2}$$

$$\mathrm{MPR}(a) = \mathrm{Max}(Z\,n\,i\,\text{number of review}()) \tag{3}$$

2) Membership Length: Eq. (4) defines it as the number of days between now and the day the reviewer account was established.

$$\mathrm{M}(a) = \text{today} - \text{yelpJoinDate} \tag{4}$$

3) Review Count: It shows the number of reviews a reviewer posts.

After creating the feature vector, PU learning is employed, resulting in two classes for the classification process. The classifiers are trained using a two-step approach of PU learning: finding the reliable negative from unlabeled instances and iteratively training SVM, Naive Bayes. The second step uses a learning algorithm to generate a two-class classifier on the refined training data. In Fig. 5, we have shown the implementation of our approach for detecting fake reviews. The method is iterative with two steps involved: A negative class is thought to apply to the whole unlabeled set. The classifier is trained using this collection and a set of positive samples. This classifier is used in step two to categorize unlabeled sets automatically labelled in step one. The cases in the unlabeled set determined to be positive are eliminated. The remainder is trustworthy negative examples to be applied in the next cycle. Until the stop benchmark is reached, this iterative method is repeated. The most recently constructed classifier is then used to determine whether a review is fraudulent or real [29].

The method is quite similar to Ott et al. [30] as this technique aims to detect deceptive and genuine reviews spontaneously. Ott et al.'s method rely on labeled negatives, which are harder to get, and real-time classification cannot be achieved through conventional text identification methods. This method can build two or more classifiers using positive and unlabeled samples. The extraction of reliable negatives is attained by automatic iterative classification, and the instances from the unlabeled sets classified as positive are removed. The remaining ones are treated as reliable negative instances as shown in Fig. 6.

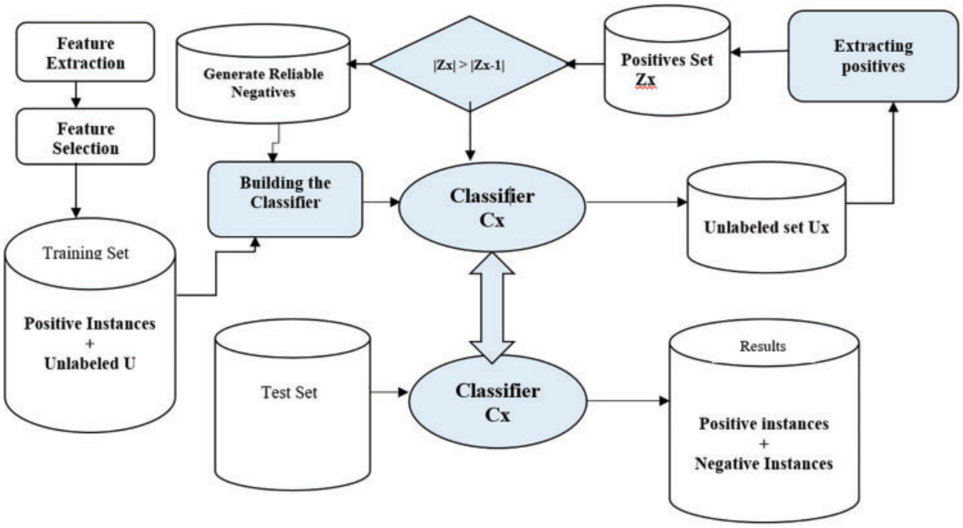


Figure 6: Generating classifier for fake reviews by utilizing PU-learning

The set of labeled and unlabeled sets is used in the training process. Positive samples and unlabeled sets first train the classifier. Then this classifier is used to label the instances of unlabeled data sets loaded during the training phase. After extraction of positive labels, they are removed, and a new unlabeled set is generated. This process is repeated until the unlabeled data set becomes smaller than the set generated in the previous iteration. The final classifier is returned after loop termination and this process labels unlabeled sets and incrementally builds the final classifier.

P = positive samples, U = Unlabeled samples at each iteration, O = the original unlabeled sample, CL = the classifier at each iteration, Z = Unlabeled instances classified as positive. Algorithm 1 is used for PU learning approach described.

Algorithm 1: Positive unlabeled learning for fake review detection

```
$x = 1$
$|Z0| = 0$
$|Z1| = 0$
While $|Zx| <= | Zx-1|$ do
CLx = Build Classifier (P, Ux)
ULx = Cx(Ux) //labelling the unlabeled set
Zx = Extract positives (ULx)
Ux+1 = Ux – Zx // Subtracting Positive set from unlabeled set to generate reliable negatives
x++
```

3.5 Classification

This study employed two classifiers, Nave Bayes and Support vector machines, to categorize reviews as bogus or legitimate. A probabilistic multi-class classification technique called Naive Bayes, which relies on the Bayes theorem, assumes that input characteristics are independent before predicting an output class. The likelihood of features belonging to a certain class—for example, LIWC, POS tagging, N-grams, and cosine similarity—is estimated using the training data. To anticipate the class testing results, generate a probability value for each attribute component of the existing classes. The root of this machine learning model is the Bayes theorem.

$$P(M|N) = P(N|M)P(M)/P(N) \tag{5}$$

The probability of A can be calculated given that B has already occurred, and B is the evidence, while A is represented as a hypothesis as shown in Eq. (5). The assumption associated with this technique is that all features are independent, they do not affect each other, and they also contribute equally. NB is commonly used for sentiment analysis, spam filtering, etc. The main advantages of this method are that it is fast, easy to implement, and suitable for multi-class prediction problems; if the assumption is true, then it requires lesser training data, thus making it suitable for PU learning and categorical input variables. SVM is a binary classification algorithm that classifies samples into one or two classes (fake or genuine). It can also perform well with minimal data, the positive instances from the data set. SVM classifies samples by discovering the extreme margin hyperplane that can categorize the cluster of text's features amongst two classes (0 or 1), the maximum space between the hyperplane and the nearest sample from either class. The advantages of SVM are: effective in high dimensional space, and uses a subset of training points, making it memory efficient and adaptable.

4 Experimental Results

This section discusses the complete results after applying unigram, bigram models, and smoothing techniques on the dataset. We perform recursive experiments to evaluate our approach's proficiency by developing custom modules on an HP Elite machine with Intel(R) Core (TM)i5-8200UCPU @ 2.80 GHz, RAM 12 GB, and operating system Windows 11 Home 64-bit. After training classifiers, the prediction achieved is analyzed through the parameters of the confusion matrix. The performance of Nave Bayes and SVM is assessed for binary classification (Fake, real) using the confusion matrix. The elements of the matrix are as follows:

True positive (TP): The total number of fake reviews classified as counterfeit by the classifier.

True negative (TN): The total number of real reviews classified as real by the classifier.

False Positives (FP): The real reviews are classified as fake.

Fall Negatives (FN): The fake reviews are classified as real.

These components are further calculated to provide insight into how well the models worked. As a result, the following performance metrics were produced from the confusion matrix's components.

Precision: Precision measures how correctly a developed TP system works for the correctness of the model. In our case, it gives the number of correctly identified named entities out of the total tagged named entities, as indicated in Eqs. (6) and (7).

$$Accuracy = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{6}$$

$$\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{7}$$

Recall: A recall computes the completeness of a model. Our system is the ratio of correctly tagged named entities to the total number of named entities in data. Recall confirms the total number of positive results retrieved but does not tell about the irrelevant results. Its Eq. (8) is shown below:

$$\mathrm{Recall/sensitivity} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{8}$$

F-Measure: F-measure can be computed using the weighted Harmonic Mean (HM) of precision and recall, as indicated in Eq. (9). Lower precision increases recall, and lower recall increases precision measure.

$$\text{F-score} = (2 \times \mathrm{Precision} \times \mathrm{Recall})/(\mathrm{Precision} + \mathrm{Recall}) \tag{9}$$

There are 1600 reviews in the data set by Ott et al. [30], 800 of which are deceptive and 800 of which are truthful. Fake and genuine reviews are differentiated by associating a binary tag with each category. Tag 0 is assigned to fake reviews and 1 denotes a real review. The data set of genuine reviews is divided into half, 400 reviews are assigned positive sentiment polarity. In contrast, the remaining ones are assigned a negative sentiment polarity and a similar technique is used for the deceptive reviews class.

Similarly, the product reviews in the data set generated by Tian et al. have fake and genuine reviews for products; we have selected 400 reviews, half of which are assigned positive polarity while the remaining ones are marked as negative. 400 deceptive and 400 truthful reviews are extracted from the entire data set to create a fixed testing set. The remaining 800 reviews are used for training the classifiers. The training sets are created by using samples of different sizes of deceptive reviews. The first performance comparison creates four training sets of 600, 800, 1000, and 1200 deceptive reviews. The remaining unlabeled data has reviews distributed as truthful and deceptive reviews. The highest F-score using Naive Bayes for truthful reviews is 0.81; for SVM, the F-score for truthful reviews reached a value of 0.73 as shown in Tables 1 and 2.

Table 1: Comparison of training sets 600, and 800 positive samples

SVM		Truthful	Precision	Recall	F-score
			0.51	0.96	0.67
Naive bayes	600-positive Samples	Deceptive	0.70	0.11	0.19
		Truthful	0.57	0.95	0.72
		Deceptive	0.91	0.26	0.41
PU leaning		Truthful	0.65	0.97	0.75
		Deceptive	0.93	0.35	0.55
SVM		Truthful	**Precision**	**Recall**	**F-score**
			0.56	0.93	0.70
Naive bayes	**800-positive Samples**	Deceptive	0.78	0.28	0.41
		Truthful	0.61	0.94	0.74
		Deceptive	0.84	0.41	0.56
PU leaning		Truthful			
		Deceptive			

Table 2: Comparison of training sets 1000 and 1200 positive samples

SVM		Truthful	Precision	Recall	F-score
			0.61	0.91	0.73
Naive bayes	1000-positive samples	Deceptive	0.83	0.41	0.55
		Truthful	0.80	0.75	0.81
		Deceptive	0.78	0.91	0.84
PU leaning		Truthful	0.85	0.94	0.83
		Deceptive	0.81	0.90	0.87
SVM		Truthful	**Precision**	**Recall**	**F-score**
			0.62	0.74	0.67
Naive bayes	**1200-positive samples**	Deceptive	0.66	0.54	0.60
		Truthful	0.71	0.86	0.78
		Deceptive	0.80	0.78	0.80
PU leaning		Truthful	0.73	0.87	0.82
		Deceptive	0.85	0.80	0.87

The following figures show these algorithms' accuracy, recall, and F1-score values. Figs. 7–9 show the proposed models' accuracy, recall, and F1-score value.

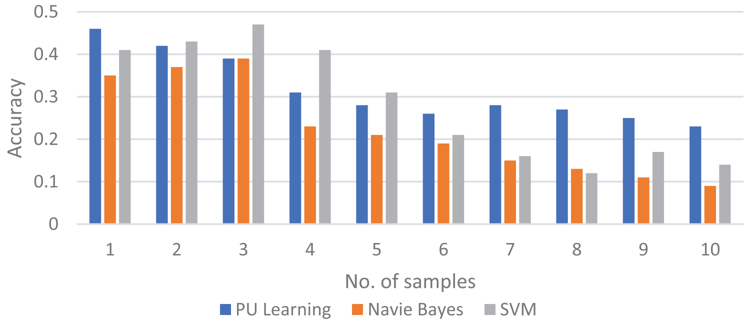


Figure 7: Comparison of accuracy of different models on different samples

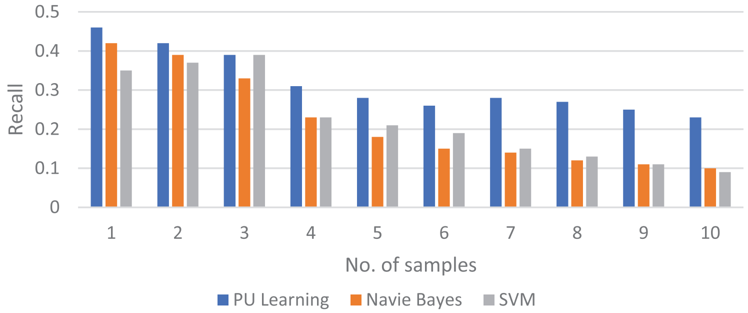


Figure 8: Comparison of recall value of different models on different samples

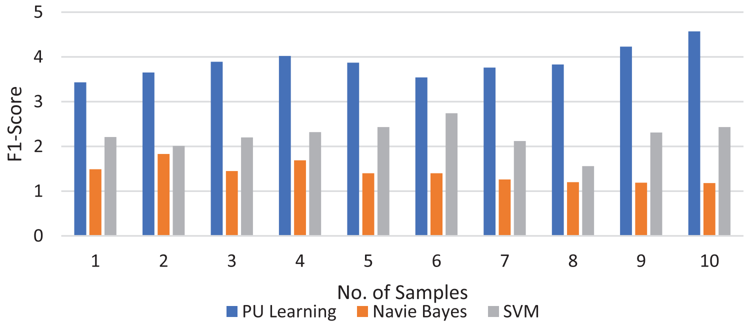


Figure 9: Comparison of F1-score value of different models on different samples

We have conducted a comparative analysis, adding the results to the paper. In Figs. 10 and 11, we have compared the results of the proposed solution with other schemes, and it shows that the proposed solution performed well when accuracy was computed. Similarly, the recall value shows that the proposed solution outperforms the existing studies.

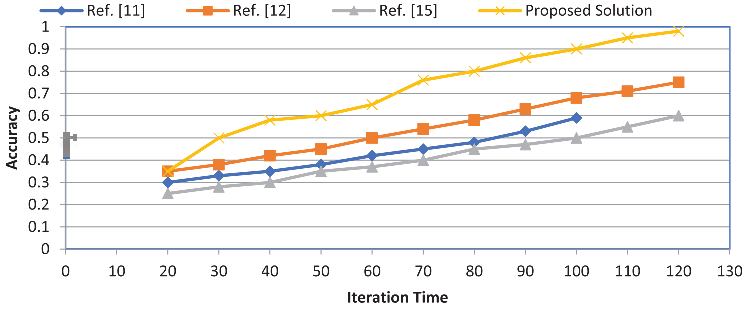


Figure 10: Comparison of existing approaches with proposed solution for accuracy

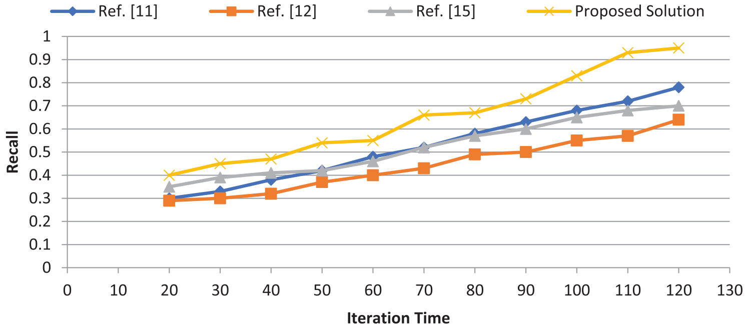


Figure 11: Comparison of recall of existing approaches with the proposed solution

4.1 Statistical Comparison

We have compared the results with other techniques and some other models and found that the proposed work significantly improves the performance of the fake review detection system. We employed two human annotators who independently solved the job while labeling the data, such as when looking for agreements for possible matches between reviews inside the fake and actual reviews dataset. We used a third human annotator to mediate disagreements in mismatches.

In addition to presenting the p (probability) value for statistical tests, we also computed the effect size. Using Cohen's d, which is the difference between means divided by the pooled standard deviation, we determined the effect size for *t*-tests. By dividing the z distribution by the sample size's square root, we could compute the r value for Wilcoxon tests. We describe the effect size according to Cohen's d (0.2 = small, 0.5 = medium, 0.8 = big) and the correlation coefficient r (0.10 = small, 0.30 = medium, 0.50 = large). Although a statistical difference was seen in two instances, the effect size showed a small magnitude of the changes. For this, extra samples (false reviews) must be included in the tests to demonstrate a statistically significant difference. Furthermore, we are unable to guarantee that all genuine reviews are authentic. As previously said, the issue we are attempting to tackle in this work is identifying and eliminating all fraudulent reviews to give a gold-standard dataset for normal reviews. Because of this, we solely focused on precision while optimizing the classifier.

4.2 Discussion of the Results

Our suggested strategy works quite well, and the results are very encouraging. We have various proofs to back up this assertion, including classification accuracy under the balancing distribution was

67.8%, which is much better than the accuracy of 50% of random guessing. This suggests considerable language differences between filtered and unfiltered evaluations, suggesting that the two reviewers have distinct psychological states when they write reviews. Second, Yelp has been filtering for a long time. Despite some accusations that real reviews are being filtered, some false positives are to be expected, given the vast majority of filtered reviews on Yelp. If Yelp's filtering was random, we think it would not have been used for the previous 6–7 years. Even if they are not concrete proof, they comfort us that Yelp is operating reasonably and that its filtering is trustworthy enough. The suggested approach is particularly suitable for opinion spam identification, according to the assessment of the method in a collection of hotel reviews. It was trained using many unlabeled samples and just 100 positive examples, and it could classify misleading views with an F-measure of 0.84.

4.3 Future Scope and Applications of Fake Review Detection

Identifying fraudulent reviews challenges internet businesses, e-commerce platforms, and academics alike. Our findings show that existing text-generation techniques produce phony evaluations that are difficult for a person to recognize because they seem so genuine. Thankfully, machine learning classifiers do much better in this aspect, identifying evaluations produced by other machines with almost flawless accuracy. This suggests that when combating bogus reviews, "machines can fight machines". Experiments with other datasets and platforms will need more investigation.

Finally, what false means for marketing is an intriguing and significant subject. What should marketers do when malevolent actors use internet platforms more often to disseminate false information using texts, audio, videos, and photos that were made artificially? While these issues are beyond the purview of this publication, we still stress the need of a continuing series of research projects addressing these issues, including how to improve both technological and non-technical defenses against counterfeit goods. Technology is an opponent as well as an ally in this endeavor. While dishonest marketers may use text-generation algorithms to generate phony evaluations on a big scale, ethical marketers can also utilize them to create countermeasures like more effective detectors that discourage dishonesty.

5 Conclusion

Consumers increasingly often utilize product reviews while choosing products. However, reviewers manipulate the system by submitting false reviews to promote or denigrate the target items because they do it for financial gain. A dataset including both negative and positive reviews is used to train a semi-supervised PU learning and machine learning algorithm to identify false reviews. A collection of unlabeled data and a few labeled positive cases may both be used in PU learning, a semi-supervised methodology. The highest F-score using Naive Bayes for truthful reviews is 0.81; for SVM, the F-score for truthful reviews reached a value of 0.73. When only deceptive labeled instances are used, the highest F-score value attained is 0.60 for SVM and 0.81 for Naive Bayes-based classification. Also, the best F-score is achieved by using 1000 samples of positive instances. The promising findings show that training with only one hundred instances of false views may lead to measures of 0.87 and 0.96 for false opinions that are positive and negative, respectively. The results consistently beat the original approach's findings in both types of false views, demonstrating the suitability of the suggested PU-learning conservative variation for opinion spam detection.

Acknowledgement: I would like to thank Prince Sattam bin Abdulaziz University for their support throughout my research.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design: A. Alshehri; data collection: A. Alshehri; analysis and interpretation of results: A. Alshehri; draft manuscript preparation: A. Alshehri. The author reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Wang and K. Y. Kevin, "Understanding the Message and formulation of fake online reviews: A language-production model perspective," *AIS Transactions on Human-Computer Interaction*, vol. 14, no. 2, pp. 207–229, 2022.

[2] J. Rout, S. Singh, S. Jena, and S. Bakshi, "Deceptive review detection using labeled and unlabeled data," *Multimed. Tools Appl.*, vol. 76, no. 3, pp. 3187–3211, 2016.

[3] D. K. Dixit, B. Amit, and D. Dharmendra, "Fake news classification using a fuzzy convolutional recurrent neural network," *Comput. Mater. Contin.*, vol. 71, no. 3, pp. 5733–5750, 2022.

[4] A. Gutub, M. K. Shambour, and M. A. Abu-Hashem, "Coronavirus impact on human feelings during 2021 hajj season via deep learning critical twitter analysis," *J. Eng. Res.*, vol. 11, no. 1, pp. 100001, 2023.

[5] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A hybrid approach for alluring ads phishing attack detection using machine learning," *Sens.*, vol. 23, no. 19, pp. 8070, 2023.

[6] A. Iqbal, R. Amin, J. Iqbal, R. Alroobaea, A. Binmahfoudh and M. Hussain, "Sentiment analysis of consumer reviews using deep learning," *Sustainability*, vol. 14, no. 17, pp. 10844, 2022.

[7] A. A. Munshi and A. Alhindi, "Big data platform for educational analytics," *IEEE Access*, vol. 9, pp. 52883–52890, 2021.

[8] G. Shan, L. Zhou, and D. Zhang, "From conflicts and confusion to doubts: Examining review inconsistency for fake review detection," *Decis. Support. Syst.*, vol. 144, pp. 113513, 2023.

[9] P. K. Jain, R. Pamula, and G. Srivastava, "A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews," *Comput. Sci. Rev.*, vol. 41, pp. 100413, 2021.

[10] A. Molla, Y. Biadgie, and K. A. Sohn, "Detecting negative deceptive opinion from tweets," in *ICMWT 2017*, Kuala Lumpur, Malaysia, Jun. 2017, pp. 329–339.

[11] D. U. Vidanagama, T. P. Silva, and A. S. Karunananda, "Deceptive consumer review detection: A survey," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 1323–1352, 2022.

[12] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection: An empirical study," *Inform. Sci.,* vol. 385, pp. 213–224, 2017.

[13] D. He *et al.*, "Fake review detection based on PU learning and behavior density," *IEEE Netw.*, vol. 34, no. 4, pp. 298–303, 2020.

[14] G. S. Budhi, R. Chiong, Z. Wang, and S. Dhakal, "Using a hybrid content-based and behaviour-based featuring approach in a parallel environment to detect fake reviews," *Electron. Commer. Res. Appl.*, vol. 47, pp. 101048, 2021.

[15] J. Fontanarava, G. Pasi, and M. Viviani, "Feature analysis for fake review detection through supervised classification," in *2017 IEEE Int. Conf. DSAA*, Tokyo, Japan, Oct. 2017, pp. 658–666.

[16] G. Su, W. Chen, and M. Xu, "Positive-unlabeled learning from imbalanced data," in *Proc. 30th IJCAI*, Montreal, Canada, Aug. 2021, pp. 2995–3001.

[17] S. X. Zhang, A. Q. Zhu, G. L. Zhu, Z. L. Wei, and K. C. Li, "Building fake review detection model based on sentiment intensity and PU learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 6926 –6939, 2023.

[18] S. N. Alsubari, S. N. Deshmukh, M. H. Al-Adhaileh, F. W. Alsaade, and T. H. Aldhyani, "Development of integrated neural network model for identification of fake reviews in E-commerce using multidomain datasets," *Appl. Bionics. Biomech.*, vol. 2021, pp. 5522574, 2021.

[19] C. Sun, Q. Du, and G. Tian, "Exploiting product-related review features for fake review detection," *Math. Probl. Eng.*, vol. 2016, 2016.

[20] E. Kauffmann, J. Peral, D. Gil, A. Ferrández, R. Sellers and H. Mora, "A framework for big data analytics in commercial social networks: A case study on sentiment analysis and fake review detection for marketing decision-making," *Ind. Mark. Manag.*, vol. 90, pp. 523–537, 2020.

[21] M. C. de Souza, B. M. Nogueira, R. G. Rossi, R. M. Marcacini, B. N. Dos Santos and S. O. Rezende, "A network-based positive and unlabeled learning approach for fake news detection," *Mach. learn.*, vol. 111, no. 10, pp. 3549–3592, 2022.

[22] C. Gong, "Large-margin label-calibrated support vector machines for positive and unlabeled learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3471–3483, 2019.

[23] P. Nair and I. Kashyap, "Hybrid pre-processing technique for handling imbalanced data and detecting outliers for KNN classifier," in *Int. Conf. COMITCon*, Faridabad, India, Feb. 2019, pp. 460–464.

[24] S. Aleem, N. U. Huda, R. Amin, S. Khalid, S. S. Alshamrani and A. Alshehri, "Machine learning algorithms for depression: Diagnosis, insights, and research directions," *Electron.*, vol. 11, no. 7, pp. 1111, 2022.

[25] S. Rao, A. K. Verma, and T. Bhatia, "A review on social spam detection: Challenges, open issues, and future directions," *Expert. Syst. Appl.*, vol. 186, pp. 115742, 2021.

[26] M. Mehmood, R. Amin, M. M. A. Muslam, J. Xie, and H. Aldabbas, "Privilege escalation attack detection and mitigation in cloud using machine learning," *IEEE Access*, vol. 11, pp. 46561– 46576, 2023.

[27] J. Koven, H. Siadati, and C. Y. Lin, "Finding valuable yelp comments by personality content geo and anomaly analysis," in *ICDM Workshop*, Shenzhen, China, Dec. 2014, pp. 1215–1218.

[28] J. Z. Wang, Z. Yan, L. T. Yang, and B. X. Huang, "An approach to rank reviews by fusing and mining opinions based on review pertinence," *Inf. Fusion*, vol. 23, pp. 3–15, 2023.

[29] Y. Zheng, H. Peng, X. Zhang, Z. Zhao, X. Gao and J. Li, "DDI-PULearn: A positive-unlabeled learning method for large-scale prediction of drug-drug interactions," *BMC Bioinform.*, vol. 20, pp. 1–12, Dec. 2019.

[30] 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 of the Assoc. Comput. Linguist.: Hum. Lang. Technol.*, Portland, OR, USA, Jun. 2011, pp. 309–319.