Fake Consumer Review Detection using Deep Neural Networks Integrating Word Embeddings and Emotion Mining

Petr Hajek[1*], Aliaksandr Barushka[2], Michal Munk[3]

[1*] corresponding author, Institute of System Engineering and Informatics,

Faculty of Economics and Administration, University of Pardubice, Studentská 84, 532 10 Pardubice,

Czech Republic

e-mail: petr.hajek@upce.cz, tel.: +420 466 036 147, fax: +420 466 036 010

[2] Institute of System Engineering and Informatics, Faculty of Economics and Administration,

University of Pardubice, Studentská 84, 532 10 Pardubice, Czech Republic,

e-mail: aliaksandr.barushka@student.upce.cz

[3] Department of Computer Science, Constantine the Philosopher University in Nitra, 949 74 Nitra, Slovakia,

e-mail: mmunk@ukf.sk

Abstract. Fake consumer review detection has attracted much interest in recent years owing to the increasing number of Internet purchases. Existing approaches to detecting fake consumer reviews use the review content, product and reviewer information and other features to detect fake reviews. However, as shown in recent studies, the semantic meaning of reviews might be particularly important for text classification. In addition, the emotions hidden in the reviews may represent another potential indicator of fake content. To improve the performance of fake review detection, here we propose two neural network models that integrate traditional bag-of-words as well as the word context and consumer emotions. Specifically, the models learn document-level representation by using three sets of features: (1) *n*-grams, (2) word embeddings and (3) various lexicon-based emotion indicators. Such a high-dimensional feature representation is used to classify fake reviews into four domains. To demonstrate the effectiveness of the presented detection systems, we compare their classification performance with several state-of-the-art methods for fake review detection. The proposed systems perform well on all datasets, irrespective of their sentiment polarity and product category.

Keywords: neural network, deep learning, fake review, review spam, word embedding, emotion

1 Introduction

Fake consumer reviews provide a fictitious and misleading opinion that does not reflect a consumer's authentic product experience. They can be submitted and published on multiple online platforms such as shopping portal forums. Globally, the number of users of these platforms has steadily increased over recent years. For example, TripAdvisor, the world's largest online travel platform, has over 455 million unique visitors in an average month and 600 million consumer reviews of 7.5 million varieties of accommodation, restaurants and attractions [68]. Generally, online product reviews provide valuable information for consumers, who increasingly rely on them and consider them to be a trusted source of information [63]. Most marketplaces prioritize well-evaluated products (the so-called snowball effect), potentially rewarding businesses paying for fake reviews. Review volume and review valence have been reported to be significant determinants of retail sales in a meta-analysis of more than 20 empirical studies [21]. This is particularly relevant for high-involvement products that can only be reviewed upon consumption. Consumers' experience of product use is therefore an important assumption. As shown in a recent survey [14], more than 80% of consumers trust online reviews as much as they trust personal recommendations. For business, it is therefore increasingly tempting to purchase fake reviews. Competitive advantage can be easily achieved by producing positive fake reviews for their products or negative fake reviews for competition products. It is also easy to find freelance writers that can produce a large number of fake reviews. Indeed, recent statistics show that every third review on TripAdvisor is fake [67]. Therefore, online reviews have become a major concern for the industry. To guarantee fair competition, it is thus important for online platforms to detect and remove fake reviews and ban fraudulent users.

Fake reviews can be identified either manually or automatically [56]. However, manual fake review detection is usually expensive, slow and relatively inaccurate compared with automatic detection approaches [26]. Over the past decade, a number of important advances have been made to improve automatic fake review detection. Machine learning approaches such as support vector machines (SVMs) and neural networks (NNs) have gained a reputation as being effective methods for fake review detection [28, 30]. Such approaches use the review content, a user's behaviour and other features to accurately classify reviews as truthful or fake. Further, it is crucial to achieve a low false positive rate because, otherwise, users of online platforms could not access truthful reviews and trustworthy users would be penalized and lose motivation to post reviews on the platform. The main idea of

machine learning approaches based on the review content is to build a list of words or phrases with weights assigned to each word or phrase (bag-of-words) or entire word categories (psycholinguistic or part-of-speech tagging) [18]. However, such word representations suffer from sparsity, making it difficult to capture the semantic representation of consumer reviews. To overcome this problem, Ren and Ji [61] developed a gated recurrent NN model combining sentence representations to detect deceptive opinion spam. Their approach adopted word embeddings learnt using the continuous bag-of-words (CBOW) model [38, 50] so that semantically similar words are mapped to word vectors based on their context. Thus, global semantic representation can be obtained, and the problem of scarce data is eliminated to a certain degree. Similarly, Li et al. [43] proposed a convolutional neural network (CNN) that combines different sentence representations into a document representation model.

Inspired by these state-of-the-art models [43, 61], here we use word embeddings to obtain the semantic representation of consumer reviews. One major drawback of the CNN model developed in [43] is that the word embeddings were trained on small datasets of several hundred reviews, which is not effective for fake consumer review detection. Indeed, word embeddings pre-trained on a large corpus achieved significantly higher accuracy in [61]. However, as reported by its authors [50], the CBOW model used in [61] is not effective in generating a generalizable context model. In contrast to [61], we employ a Skip-Gram Word2Vec model to produce word embeddings from a corpus of consumer reviews. Compared with the CBOW model, the Skip-Gram model exploits the word context more effectively [50]. Another limitation of the above deep NN models is that only word embeddings have been considered, ignoring emotion indicators of consumer reviews. As reported in [59], review ratings are rarely consistent with the sentiment polarity of the review content. Moreover, the sentiment strength can be substantially different for reviews with the same rating scores. Most importantly, the sentiment strength has been proven to be significantly more effective than ratings in detecting fake reviews [59]. To overcome the problems of existing deep NN models and further improve detection performance, we combine the produced word embeddings with bag-of-words and several lexicon-based emotion indicators. The latter have been limited to positive and negative sentiment in recent studies [5, 35, 47]. However, additional emotion categories such as trust and aptitude have been reported to be effective indicators of online review helpfulness [20, 49]. Here, we incorporate those emotion indicators into the fake review detection model using deep learning.

Deep NNs have successfully been used for related spam detection tasks in recent studies, including email spam detection [6, 8] and spam detection in social media [7, 32, 33, 48]. In this study, two deep NN models, namely a deep feed-forward neural network (DFFNN) and a CNN, are used to capture the complex features hidden in high-dimensional word, sentence and emotion representations. In summary, the contributions of this study are as follows:

- Developing a novel fake review detection model integrating word, sentence and emotion representation. The novelty of our model lies in the effective exploitation of the word context in consumer reviews considering bag-of-words and emotion representations. This is also the first study using different emotion representations for fake review detection.
- Using various benchmark datasets for a wide range of consumer products and services, showing that the proposed approach can have higher classification performance than state-of-the-art fake review detection methods.

This study is a significantly extended version of [10]. The earlier version was limited to the Skip-Gram model trained on a small corpus of hotel reviews, without considering emotion mining from reviews. Furthermore, here we propose an improved DFFNN model and a novel CNN-based fake detection model and examine the effect of different word, sentence and emotion representations on fake review detection performance. The extension further includes an in-depth comparative analysis with state-of-the-art fake detection methods on several datasets.

The remainder of this paper is organized as follows. Section 2 reviews related work on detecting fake reviews. Section 3 presents the four datasets used for the experimental comparison. Section 4 outlines the proposed model for fake review detection. Section 5 presents the experimental results and a comparative analysis with several state-of-the-art methods used for fake review detection. The final section concludes and suggests possible future directions.

2 Related Work

Fake reviews have increasingly been recognized as a major concern for online shopping. To affect consumers' decisions and thus achieve competitive advantage, positive and negative fake reviews are intended to promote or

demote target products [61]. As consumers have limited capacity to identify fake reviews [26, 28], machine learning methods have been employed to ensure their early detection. To automatically classify reviews into fake or truthful classes, an annotated corpus of reviews (with class labels) is typically used for training and testing. A considerable number of studies have been published on the automatic detection of fake reviews in the past decade. The list of those studies in Table 1 presents the features and methods used, datasets and resulting performance evaluation.

Jindal and Liu [34] presented the first study aimed at detecting fake product reviews based on the similarity of review and product features. More precisely, spammers' tendency to duplicate their product reviews was used. Motivated by this early effort, the studies that followed [41, 44] developed fake review detection systems using the cosine similarity between reviews. To detect spammers who can adapt their behaviour, Wang et al. [75] proposed a heterogeneous review graph that captures the relationships among reviews, reviewers and reviewed shops. Thus, the trustiness of reviewers, honesty of reviews and reliability of shops could be calculated without considering the review content. Inspired by this approach, Liu et al. [47] proposed a probabilistic graph classifier in which the multimodal embedded representation of nodes is obtained using a bidirectional NN with an attention mechanism. By contrast, Lau et al. [37] developed a fake review detection approach based on text mining only. Several types of features were used in [41], including the review content as well as its sentiment, product features and user profile, to classify fake reviews using semi-supervised machine learning methods. Review metadata (content, timestamp and rating) were then combined with relational data in a unified semi-supervised framework called SpEagle [60]. Ghai et al. [24] showed that the rating deviation of a particular review from others indicates fake reviews. Spam attacks were reported to be correlated with review ratings and, therefore, abnormal temporal patterns in the ratings may indicate spam attacks [71]. By elaborating on this idea, a list of indicative signals of fake reviews over time was used for the real-time detection of abnormal review events [40, 73]. Furthermore, temporal features were combined with users' spatial patterns to find that fake reviews exhibit geographical outsourcing and that fake users are more active on weekdays [39]. A rule-based feature weighting scheme was proposed in [3] to combine review-based, reviewer-based and product-based features.

Most fake review detection systems extract informative features from the review content. Such features are typically represented by bag-of-words (*n*-grams) [58, 59], psycholinguistic word lists (e.g., positive/negative words or

spatial words) [42] and part-of-speech tagging (e.g., first-person pronouns) [43]. Aspect sentiment was used in [46] to detect fraudulent users. Xue et al. [72] integrated the deviation of a user's aspect sentiment into a framework to calculate the trust scores for users, reviews and products. Further, word embeddings have recently been used to obtain the semantic representation of reviews. In [61], the pre-trained CBOW model was tuned on actual review datasets using CNN to improve detection accuracy. The CBOW model was also used together with relational features to develop a semi-supervised framework in [74]. Moreover, word embeddings were trained using sentence-based CNNs to produce document representations for fake review detection in several product domains [43].

Table 1: Summary of previous studies on fake review detection

Study	Content-based features	Classifier	Dataset	Performance
[34]	positive/negative words, brand name, similarity of review and product features, numeric and capital words	LR	Amazon	AUC=0.78
[41]	unigrams and bigrams, review length, first-person pronouns, similarity with other reviews, ratio of question sentences, ratio of the capital letters, subjective/objective words, positive/negative words	NB, Co-training	Epinions	*F*-score=0.63
[15]	user rating, app rating	DT, LCGM	App Store	
[56]	unigrams and bigrams	SVM	Hotels	Acc=0.86
[64]	frequency of characters, words and punctuation marks	SVM	Hotels	*F*-score=0.84
[52]	unigrams and bigrams	SVM	Yelp	Acc=0.86
[42]	unigrams, positive/negative words, spatial words, first-person pronouns	SAGE	Hotels and doctors	Acc=0.65
[39]	unigrams and bigrams	SVM	Restaurants	Acc=0.85
[60]	review length, content similarity among user's (product's) reviews	SSL	Yelp	AUC=0.79
[65]	product word embeddings, bigrams and trigrams	Bagging	Hotels, restaurants and doctors	*F*-score=0.77
[43]	sentence weights, POS, first-person pronouns	CNN, SWNN	Hotels, restaurants and doctors	Acc=0.84
[61]	CBOW word embeddings	CNN, GRNN	Hotels, restaurants and doctors	Acc=0.84
[19]	unigrams	*k*-NN, NB, DT, SVM	Movies	Acc=0.82
[63]	bigrams, LIWC, POS	*k*-NN, RF	Hotels	Acc=0.77
[74]	CBOW word embeddings	SSL	Yelp	AUC=0.83
[2]	unigrams, bigrams, trigrams and fourgrams	SVM	Hotels	Acc=0.90
[77]	first and last sentence, middle context	LSTM ensemble	Hotels, restaurants and doctors	Acc=0.83
[5]	positive/negative words, bigrams, LDA	AdaBoost	Yelp	*F*-score=0.81
[35]	Skip-Gram word embeddings, review length, capitalized words, numerals, POS, positive/negative words	BERT	Hotels, Yelp	Acc=0.89
[47]	positive/negative words, review length, first-person pronouns, multimodal embeddings	LR	Dianping	*F*-score=0.81

[10]	Skip-Gram word embeddings, unigrams, bigrams and trigrams	DFFNN	Hotels	Acc=0.89
This study	Skip-Gram word embeddings, unigrams, bigrams and trigrams, lexicon-based emotions	DFFNN, CNN	Hotels, restaurants, doctors, Amazon	

Legend: Acc – accuracy, AUC – area under ROC curve, BERT – bidirectional encoder representations from transformers, CNN – convolutional neural network, DFFNN – deep feed-forward neural network, DT – decision tree, FNR – false negative rate, FPR – false positive rate, GRNN – general regression neural network, *k*-NN – k-nearest neighbour, LCGM – latent class graphical model, LDA – latent dirichlet allocation, LIWC – linguistic inquiry and word count, LR – logistic regression, LSTM – long short term memory, NB – Naïve Bayes, POS – part-of-speech tagging, RF – random forest, SAGE – sparse additive generative model, SSL – semi-supervised learning, SVM – support vector machine, SWNN – sentence weighted neural network.

Regarding the classification methods used to detect fake and truthful reviews, machine learning methods dominated earlier research. Logistic regression was first employed as a traditional machine learning method owing to its capacity to produce a probability estimate reflecting the likelihood that a review is fake [34]. However, traditional machine learning methods such as logistic regression and *k*-NN (*k*-nearest neighbour) suffer from at least two drawbacks [8]. First, these methods are ineffective in handling high-dimensional fake review data. This is important because a large number of word features are usually present in these data. Second, they cannot deal with data sparsity effectively. This is critical because each review usually contains only a small number of words or phrases. To overcome these problems, other machine learning methods have become popular for fake review detection such as Naïve Bayes (NB) [41] and SVMs [39, 52]. Similarly, evolutionary algorithms [57] and ensemble learning methods [5, 62] have been used to overcome the problems of convergence and overfitting, respectively. The traditional machine learning methods used to detect fake reviews have been surveyed comprehensively [18, 58, 69].

Recent advances in automatic fake review detection suggest that more complex features can be extracted from high-dimensional data using deep NNs. Therefore, deep NN models such as general regression neural networks [61], generative adversarial networks [66], CNNs [43], DFFNNs [10] and long short term memory [77] have gained much attention in recent years.

As mentioned above, fake users usually do their best to make fake reviews look as trustworthy as possible. Hence, it is difficult to collect a reliable dataset of annotated (labelled) reviews. Initially, fake reviews were identified as duplicates from the same or different users on the same or different products [34]. In practice, however, the manual annotation of reviews is a time-consuming task. Li et al. [41] used review helpfulness to make manual annotation more effective. To overcome the problem of using these heuristic methods, Ott et al. [54] made the first effort to

collate a fake review dataset with gold-standard fake reviews. These fake reviews were obtained from the Amazon Mechanical Turk, a crowdsourcing service for anonymous online workers. A pool of 400 human-intelligence tasks was created to collect 400 unique fake reviews on popular hotels in the Chicago area. Finally, 400 truthful reviews were selected to match the document lengths of the fake reviews. A similar strategy was later used to create datasets for restaurants and doctors [42]. Alternatively, the filtering algorithms of Yelp [60] and Amazon [23] were used to label fake reviews. The main advantage of this approach is that those anti-fraud filters classify fake reviews quickly and accurately [60] and, thus, larger datasets can be collected.

In summary, earlier studies attempted to use content-based features to produce accurate document (review) representation. However, such representation can be complex and high-dimensional, which may result in the poor convergence of classifiers and overfitting risk. To extract higher-order features from content-based features, deep NNs have recently been employed, which can capture higher complexity and abstraction. Compared with previous approaches and to further improve the performance of fake review detection, we propose DFFNN and CNN models exploiting word embeddings (obtained using the Skip-Gram Word2Vec model pre-trained on a large corpus of consumer reviews) together with bag-of-words and emotion representations. Richer sentence and document representations of consumer reviews are produced by the proposed models compared with those mentioned above. To demonstrate the effectiveness of these models, four datasets on several consumer products and services are used in this study.

3 Datasets

We used four benchmark datasets, namely the hotel, restaurant, doctor and Amazon datasets.

The hotel dataset consists of two datasets from Cornell University[1], thus merging the positive review data [55] and negative review data [56]. Since the details of these datasets can be found in [55, 56], we only briefly describe them in this study. They were chosen because they are considered to be gold-standard fake review datasets [56]. The fake reviews were generated by unique anonymous online workers (Turkers) pretending to be customers. Only a single review per Turker was allowed and unreasonably short or plagiarized reviews were rejected. More precisely, Turkers were asked to follow these instructions: (1) they were assumed to work for the hotel's marketing

[1] See http://myleott.com/op-spam.html

department and write fake reviews as if they were customers and (2) the review was required to sound realistic and positive/negative. The corresponding truthful reviews were obtained from several online review communities such as Expedia, TripAdvisor and Hotels.com. The dataset includes 800 truthful reviews and 800 fake reviews on 20 hotels from TripAdvisor with 40 truthful and 40 fake reviews for each hotel. The positive and negative reviews were sorted, with 800 positive reviews and 800 negative reviews in the dataset. As a result, each review in the dataset has the truthful/fake label, hotel information, travel agency name, polarity (positive/negative) and review content. The reviews contained 152 words on average.

By using the same rules as for the hotel dataset, two other datasets were created by [42], one for restaurants and another for doctors. Again, well-rated US Turkers were asked to produce fake reviews. Twenty positive fake reviews were gathered for each of the 10 most popular restaurants in Chicago (i.e., 200 fake reviews in total). Similarly, 356 positive fake reviews were collected for the doctor domain. The matching sets of truthful reviews were obtained from customers, 200 for both the restaurant and the doctor domains.

The Amazon dataset[2] consists of 21,000 reviews, 10,500 of them labelled by Amazon as fake. In addition to the class label, the dataset contains a set of features for each review, including rating, verified purchase (yes or no), product category and product ID. The average rating of the reviews was 4.13 (five stars was the maximum) and 55.7% of the data was identified as verified purchases. The reviews are equally distributed across 30 distinct product categories (e.g., apparel, automotive, baby), with each category made up of 700 reviews. The product categories are identified as noncompliant with Amazon policies.

Table 2 Fake review datasets used in this study

Dataset	# fake / truthful reviews	Polarity	Aver. review length (words)
Hotel [55, 56]	400 / 400	positive and negative	151.9
Restaurant [42]	200 / 200	positive	137.1
Doctor [42]	356 / 200	positive	102.4
Amazon [23]	10,500 / 10,500	positive and negative	86.5

[2] See https://www.kaggle.com/lievgarcia/amazon-reviews

4 Methodology

In this section, we present the components of the proposed fake review detection systems, including the bag-of-words, emotion and embedding representations. Further, this section presents the DFFNN and CNN models, which are proposed to integrate those components.

4.1 Bag-of-words representation

To produce bag-of-word features, the content of the reviews was first pre-processed. To reduce noise in the dataset, we removed stopwords using the list based on Rainbow. Furthermore, the reviews' content was lowercased and special symbols were stripped out. Then, tokenization was conducted using an *n*-gram tokenizer. Unigrams, bi-grams and trigrams were selected based on the following *tf.idf* weighting scheme:

$$v_{ij} = (1 + \log(tf_{ij})) \times \log(N/df_i), \quad (1)$$

where v_{ij} is the weight of the *i*-th term in the *j*-th document, $j=1, 2, \dots, N$, tf_{ij} denotes the term frequency and df_i represents the document frequency. In agreement with previous studies [25, 31], the top 2,000 *n*-grams were selected according to their weights. Although word order and grammar were disregarded, multiplicity was retained in the *n*-gram model.

4.2 Emotion representation

As reported in [59], the review content is more important than the review rating to readers. Although it is more challenging to fake the sentiment strength in the review content than in the rating score, fake reviews are often produced by experienced professionals, which makes detecting emotion fraud a difficult task [16]. Therefore, it is important to incorporate additional product and behavioural characteristics [16]. Indeed, a similar review content for different products or the unusual time of review publication may indicate emotion fraud. In addition, it is recommended to cover a wide range of lexicon-based emotion indicators simultaneously to make the emotion analysis more reliable [12]. Compared with machine learning approaches, lexicon-based emotion indicators are less susceptible to indirect indicators of sentiment that may generate fake sentiment patterns [1].

Three types of lexicon-based emotion indicators were used in this study, including polarity, strength and emotion features [12]. Polarity-based features were represented by the numbers of positive/negative words that match two

popular lexicons, the OpinionFinder lexicon [12] and Bing Liu's opinion lexicon [29]. These lexicons were selected because they are reliable resources of handcrafted dictionaries (positive and negative word lists) with sentiment values assigned by multiple human judgements [11]. To obtain the polarity-based features, we calculated the numbers of positive and negative words for both lexicons.

To estimate the strength, we employed the following resources: (1) AFINN [53], (2) S140 [36], (3) SentiWordNet [4], (4) NRC Hashtag [36] and (5) Emoticons [53]. These lexicons provide the intensity levels of positive and negative sentiments. Five different lexicons were used due to their high level of uniqueness and neutrality [12]. Thus, their combination provided us with high lexical coverage. Unlike the polarity-oriented lexicons, the strength-based lexicons were created in semi-supervised or supervised modes from a large number of positive/neutral/negative text documents such as tweets. For each of the five lexicons, we calculated the positive and negative scores.

The emotion-based lexicons included human-provided word lists along with their corresponding tags. Eight emotions were considered in the NRC lexicon [51] based on the Plutchik wheel of emotions, namely joy, trust, sadness, anger, surprise, fear, anticipation and disgust. In addition, we used the updated version of these word lists called NRC expanded [11], which adds the emotion associations obtained from social media content. The emotion-based features were represented by the numbers of words that match those word lists.

Overall, 30 emotion indicators were obtained (4 polarity-based, 10 strength-based and 16 emotion-based features).

4.3 Word embeddings

To create word embeddings, we used the Skip-Gram model [38, 50] trained on a large corpus of ~84 million Amazon reviews[3] [27]. This model maps the words (phrases) from the vocabulary to numerical vectors to ensure that semantic similarity in the word representation is retained. To learn the Skip-Gram, a training dataset was first created from the sequences of words $w_1, w_2, \dots, w_T$. Then, the classifier's parameters and embedding function were adapted. The embedding function was applied to each word w_t in the vocabulary to produce high-dimensional word representation. Specifically, the model aimed to obtain the word representation that can predict the context words in a sentence. The objective function of the model is given as follows:

$$E = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c}\log p\left(w_{t+j}\middle|w_t\right), \quad (2)$$

[3] See http://jmcauley.ucsd.edu/data/amazon/

where c denotes the size of the window (context) and $p(w_{t+1}|w_t)$ is represented by the following hierarchical softmax algorithm [50]:

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma([\![n(w,j+1) = \mathrm{ch}(n(w,j)]\!] v'^{T}_{n(w,j)} v_{w_I}), \quad (3)$$

where w_I is the input word, v_w and v'_w are the input and output vector representations of word w, respectively, $L(w)$ is the path length from the root node to word w, $n(w,j)$ denotes the j-th node in the binary tree, $\sigma(x)$ is a sigmoidal function, ch(n) is a child node of n chosen arbitrarily, and $[\![x]\!]$=1 if x is true and otherwise $[\![x]\!]$=-1. The hierarchical softmax algorithm was used due to its computational effectiveness compared with the original softmax.

After testing the different dimensionalities of word embeddings (50, 100, 200 and 400 word vectors), we set it to k=100, which performed the best for all the tested datasets in terms of DFFNN and CNN accuracy. This setting is also in agreement with related studies [61]. The size of the context was c=5 [50] to produce a complex word representation.

4.4 DFFNN model

The DFFNN model proposed in this study was represented by a multilayer perceptron NN with two hidden layers (Fig. 1). DFFNNs can effectively process complex sparse representations of text documents just like consumer reviews [8].

In the input layer of the proposed DFFNN model, three sets of features were extracted from the raw review text, namely (1) the top 2,000 unigrams, bigrams and trigrams according to their *tf.idf* weights, (2) 30 emotion features, and (3) average embeddings calculated for each review from the pre-trained embedding weight matrix (lookup table). Additional review-based, reviewer-based and product-based features were used as inputs in this study depending on the dataset domain.

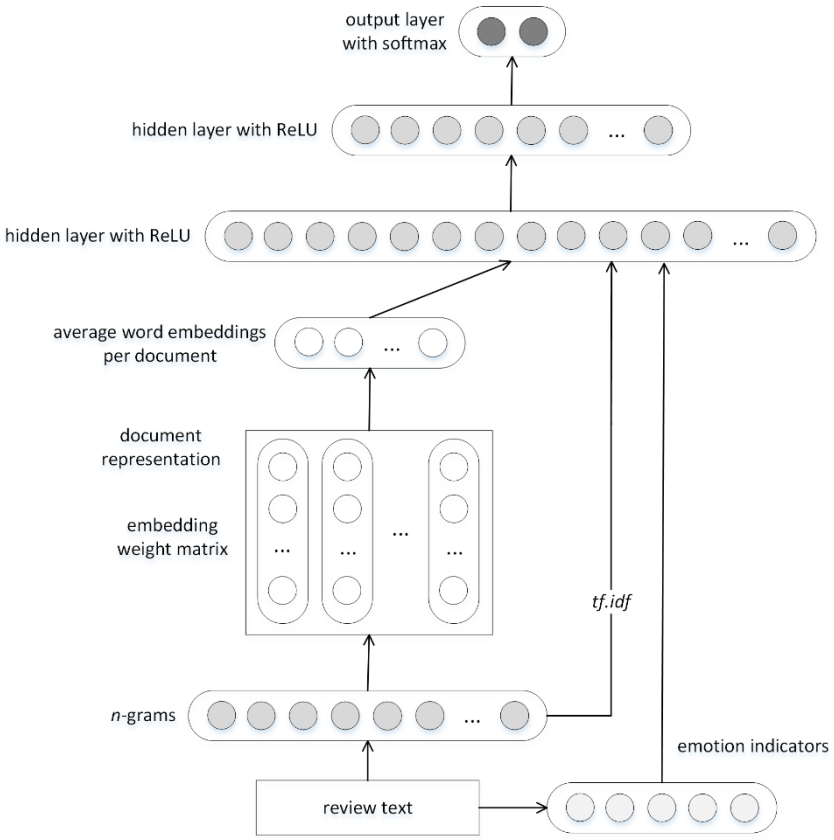

Fig. 1: DFFNN model for fake review detection

The two hidden layers process the complex relations between the input features and output classes (fake/truthful). Dropout regularization was employed during DFFNN training to avoid overfitting. In agreement with [6], rectified linear units were used in both the hidden layers to speed up the training process and avoid poor local error minima. To train the DFFNN, we used the mini-batch gradient descent algorithm that ensures stable convergence using mini-batches. The synapse weights were updated in the following way:

$$s_{t+1} = s_t - \alpha \nabla_\theta J(w_t; x^{(i:i+b)}; y^{(i:i+b)}), \quad (4)$$

where s represents the synapse weight, t is the index of iteration, t=1, 2, …, T, α denotes the learning rate, J is an objective function, x^i and y^i are the inputs and output for the i-th training sample, respectively and b is the mini-batch size. To find the optimal DFFNN structure, a grid search procedure was used for the different numbers of neurons in the hidden layers = {10, 20, 50, 100}. The dropout rate for the input layer was set to 0.2, while it was 0.5 for the hidden layers. The remaining training parameters were set as follows: b=100, η=0.1 and T=1000. In the output layer, the following softmax function was used:

$$P(y_j) = \frac{e^{\theta j}}{\sum_{k=1}^{K} e^{\theta k}}, \quad (5)$$

where θ is the set of model parameters, and j and k denote the indexes of classes. Cross-entropy loss was used to represent objective function J.

The time complexity of the proposed DFFNN model is $O(n_b \times T \times (m \times n_1 + n_1 \times n_2 + n_2 \times n_3))$, where n_b is the number of mini-batches, m is the number of features, n_1 and n_2 are the numbers of neurons in the first and second hidden layer, respectively and n_3 is the number of neurons in the output layer.

4.4 CNN model

Fig. 2 illustrates the proposed CNN architecture. For the CNN model, each sentence was converted into the k-dimensional word representation using the pre-trained embeddings, where k is the number of embeddings. Thus, the $n_w \times k$ word representation was produced, where n_w is the number words in a sentence. To obtain a static size of the inputs, the maximum n_w was set to 50. Additionally, the *tf.idf* weight and 30 emotion indicators were calculated for each word. Hence, we obtained 50×131 word representations for each sentence. Given the maximum number of sentences in the corpus of reviews, multiple word representations were created by applying these procedures to all the sentences in a review.

For the convolutional layer, we set the number of filters (feature maps) to 2^7. Again, we conducted extensive experiments with the number of filters (2^5, 2^6, 2^7, 2^8) and the additional convolutional layer, but CNN performance was not improved using those settings. A standard filter size of 5 and rectified linear unit activation was used for the convolutional layer, which was followed by a max pooling layer (size of 4) and an output layer with a softmax function. The stochastic gradient descent was used for training the CNN model with η=0.1 and T=1000.

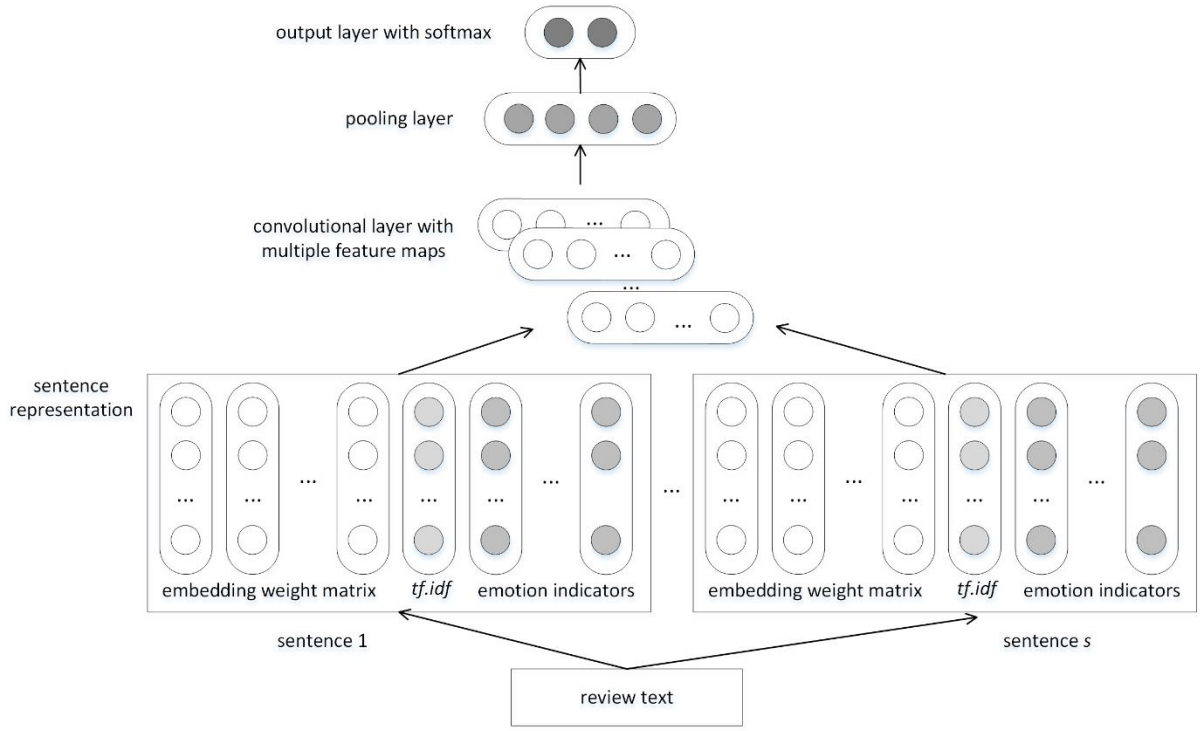

Fig. 2: CNN model for fake review detection

The time complexity of the CNN model is $O(n \times T \times (m \times s \times n_1 + n_1 \times n_2))$, where n is the number of samples, s is the number of sentences in the document, n_1 is the feature map dimension in the convolutional layer and n_2 is the number of neurons in the output layer.

5 Experimental Results

In this section, we present the results of the experiments performed to empirically evaluate the effectiveness of the proposed DFFNN and CNN models on the four fake review datasets. To evaluate the experimental results, three evaluation measures were considered: accuracy, area under the ROC (receiver operating characteristic) curve (AUC) and *F*-score. Accuracy is the percentage of reviews correctly predicted, AUC represents the probability that the classifier ranks a randomly chosen truthful review higher than a randomly chosen fake review, and *F*-score is the combination of precision (percentage of reviews correctly classified as truthful of all the reviews predicted as truthful) and recall (percentage of reviews correctly classified as fake of all the fake reviews). To consider both

accuracy and total execution time, the adjusted ratio of ratios (ARR) was also used [13]. The ARR can be defined as follows:

$$ARR_{a_p,a_q}^{d_i} = \frac{\frac{Acc_{a_p}^{d_i}}{Acc_{a_q}^{d_i}}}{1+Imp\times\log\left(\frac{T_{a_p}^{d_i}}{T_{a_q}^{d_i}}\right)}, \quad (6)$$

where $Acc_{a_p}^{d_i}$ and $T_{a_p}^{d_i}$ denote the accuracy and execution time of algorithm a_p on dataset d_i, respectively and *Imp* is the relative importance of *Acc* and *T*, indicating the amount of accuracy the user is willing to trade for execution time speedup/slowdown. Following [13], we tested three scenarios, *Imp*={0.1%, 1%, 10%}.

Hereafter, we present the means and standard deviations of the stratified 10-fold cross-validation performed on the four datasets. The Skip-Gram model, as well as the experiments with the DFFNN and CNN models were conducted in the Deeplearning4j program environment.

In the first set of experiments, we investigated the effect of word embedding pre-training on the accuracy of the DFFNN and CNN. In agreement with the results of Ren and Ji [61], we find that pre-trained word embeddings (using the corpus of ~84 million Amazon reviews) provide higher accuracy than those trained on each of the four datasets (Fig. 3). As expected, the effect was larger for the smaller datasets (restaurant and doctor). By contrast, the effect was insignificant (using the Wilcoxon signed rank test at the p=0.05 level) for the Amazon dataset, indicating that the size of this dataset was large enough to provide a reliable word representation model.

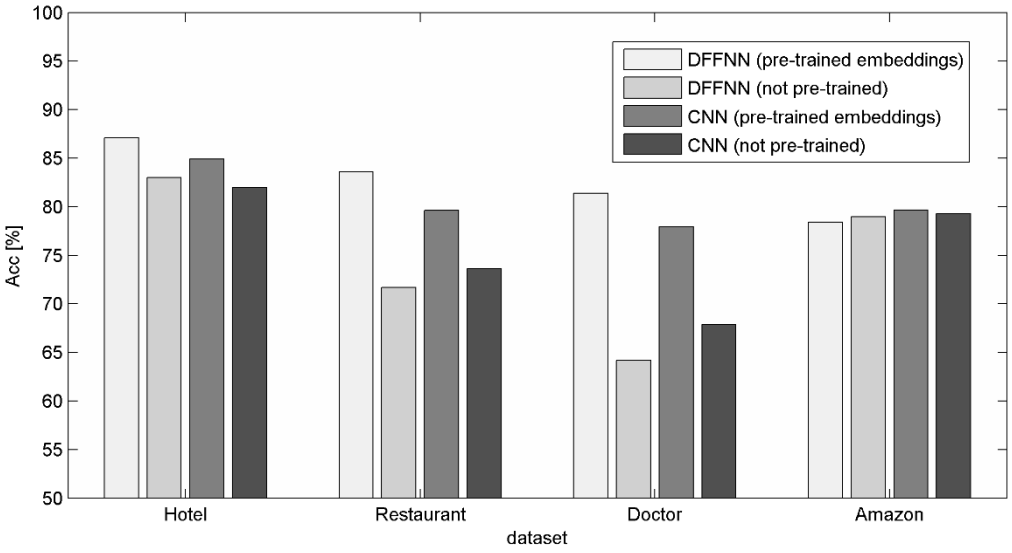

Fig. 3: The effect of pre-trained word embeddings on classification accuracy

In the further sets of experiments, we examined the effect of combining the pre-trained word embeddings (Skip-Gram model) with the *n*-gram and emotion representations (Fig. 4). The results show that in most cases, the *n*-gram model provided higher accuracy than the Skip-Gram model, suggesting that word frequency is an important indicator of fake reviews. We also tested unigrams and bigrams in the *n*-gram model but without improvement. Further improvement in accuracy was achieved using the models integrating the *n*-gram, Skip-Gram and emotion representations. Compared with the *n*-gram models, significantly higher accuracy at p=0.05 was achieved for most models, except for the restaurant and doctor datasets trained using the DFFNN. Therefore, the results indicate that the proposed integrated model was the most effective when applied to larger datasets containing reviews with both positive and negative polarity.

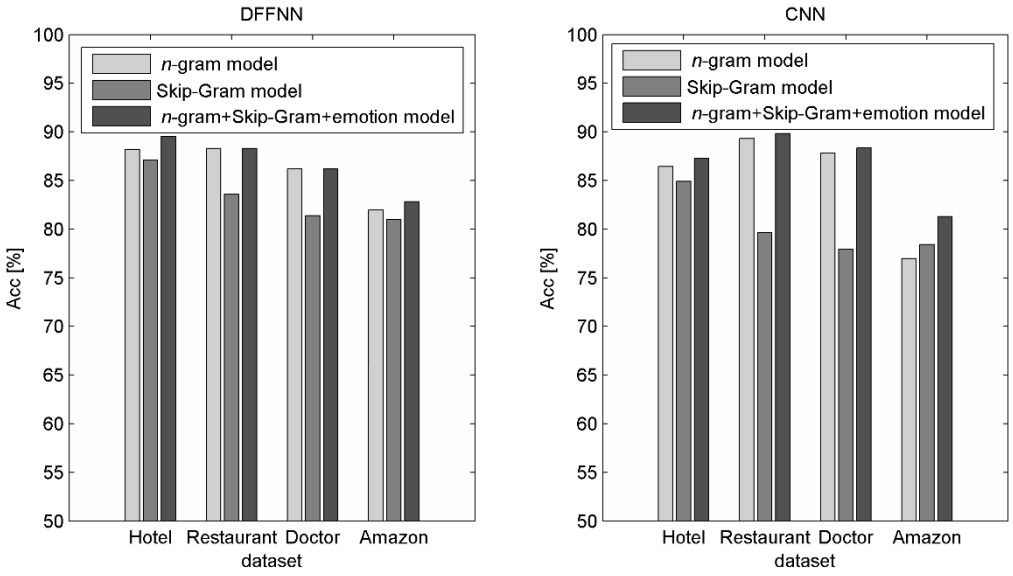

Fig. 4: The effect of combining pre-trained word embeddings on classification accuracy

To demonstrate the effectiveness of the proposed fake detection models, we also compared their performance with several baseline and state-of-the-art approaches used in previous studies of fake review detection:

- DFFNN ($DFFNN_{ngram}$) [10] and sentence CNN (SCNN) [43] models trained using *n*-grams as inputs. The NN's structures and training parameters were the same as for the proposed models (this also applies to all the NN models listed below);
- A DFFNN extension using the Skip-Gram model with 500 not pre-trained word embeddings ($DFFNN_{skipgram}$) [10];
- A CNN model using the pre-trained CBOW model with 100 pre-trained word embeddings (CNN_{cbow}) [61];
- SVM as a popular baseline model used in several related studies [26, 40, 55, 56]. The LibLINEAR implementation of the L2-regularized L2-loss SVM was used with the polynomial kernel function and varying complexity parameter $C = \{2^0, 2^1, \dots, 2^6\}$ (the optimal value of C for each dataset was determined using the grid search method);
- The following baseline models were used in earlier research: Naïve Bayes (NB) [41], Bagging [65], *k*-NN [19, 62], AdaBoost [5] and Random Forest (RF) [62]. Here, we used *k*-NN with k=3, RF with 100 random trees, and Bagging and AdaBoost M1 with REPTree and decision stump as the base learners, respectively. The experiments with these baseline models were conducted in the Weka 3.8 program environment. All the baseline models were trained using all the input features (*n*-grams, pre-trained word embeddings and emotion indicators).

Table 3 and Table 4 summarize the results of the experiments for all the compared methods in terms of Acc, AUC and *F*-score. The results are consistent across all these evaluation measures, indicating that the classifiers performed well for both, fake and truthful classes. Overall, the NN methods significantly outperformed the baseline methods (using the Wilcoxon signed rank test at the p=0.05 level), except for $DFFNN_{skipgram}$, suggesting that the latter methods have limited ability to capture complex features from the high-dimensional fake review datasets.

Table 3: Results of the experiments for the hotel and restaurant datasets

	Hotel dataset			Restaurant dataset		
Baseline methods	Acc [%]	AUC	*F*-score	Acc [%]	AUC	*F*-score
SVM[a]	80.75±3.12	0.807±0.031	0.808±0.031	80.34±7.22	0.803±0.090	0.809±0.069
NB	81.25±3.29	0.850±0.042	0.817±0.031	80.58±3.38	0.832±0.042	0.813±0.031
Bagging	78.19±4.90	0.857±0.041	0.781±0.050	77.09±6.68	0.828±0.061	0.766±0.069

k-NN	71.38±2.99	0.772±0.031	0.678±0.047	72.14±6.93	0.788±0.074	0.692±0.093
AdaBoost	77.06±2.38	0.842±0.028	0.771±0.027	74.38±6.64	0.837±0.063	0.749±0.060
RF	79.31±2.91	0.873±0.027	0.798±0.028	76.62±3.92	0.861±0.037	0.770±0.040
NN methods	Acc [%]	AUC	*F*-score	Acc [%]	AUC	*F*-score
$DFFNN_{ngram}$ [10]	88.19±2.15	**0.951±0.014**	0.882±0.022	88.31±3.91	0.938±0.038	0.887±0.035
SCNN [43]	86.44±2.41	0.939±0.020	0.863±0.023	89.30±5.76	0.952±0.041	0.898±0.054
$DFFNN_{skipgram}$ [10]	83.00±4.06	0.908±0.025	0.831±0.042	71.67±7.14	0.788±0.058	0.709±0.081
CNN_{cbow} [61]	84.88±3.25	0.911±0.026	0.850±0.032	79.61±7.86	0.889±0.055	0.803±0.064
DFFNN (this study)	**89.56±3.01**	**0.951±0.018**	**0.896±0.029**	88.31±4.71	0.953±0.030	0.884±0.047
CNN (this study)	87.25±1.70	0.945±0.014	0.872±0.015	**89.80±6.16**	**0.965±0.028**	**0.901±0.057**

Note: the best results are in bold, [a] obtained for $C=2^4$

The DFFNN model performed best for all the performance measures for the hotel and doctor datasets, while the CNN model outperformed the remaining methods for the restaurant and doctor datasets. Besides the CNN model, SCNN also performed well on those datasets, suggesting that CNNs perform better when trained on data with a single sentiment polarity. By contrast, the DFFNN model seemed to be more effective when dealing with combined sentiment (positive and negative), which can be attributed to the use of dropout in between the DFFNN layers. Indeed, this regularization approach has been reported to improve the generalization ability of deep NN models for text classification tasks [70]. The poor performance of $DFFNN_{skipgram}$ compared with CNN_{cbow} provided additional support to the effectiveness of pre-trained word embeddings. The *k*-NN and AdaBoost baseline methods also performed poorly, which can be attributed to their poor ability to handle high-dimensional datasets [8].

Table 4: Results of the experiments for the doctor and Amazon datasets

	Doctor dataset			Amazon dataset		
Baseline methods	Acc [%]	AUC	*F*-score	Acc [%]	AUC	*F*-score
SVM[a]	85.31±6.65	0.838±0.071	0.886±0.053	76.25±3.85	0.762±0.038	0.760±0.029
NB	81.02±3.90	0.827±0.054	0.853±0.028	59.21±0.92	0.633±0.012	0.617±0.009
Bagging	70.40±7.72	0.752±0.106	0.792±0.051	80.41±0.58	0.861±0.007	0.793±0.006
k-NN	71.13±3.85	0.716±0.049	0.786±0.025	75.97±0.83	0.804±0.009	0.756±0.009
AdaBoost	69.73±5.71	0.726±0.054	0.774±0.049	79.22±0.82	0.846±0.010	0.782±0.009
RF	75.07±5.59	0.812±0.061	0.831±0.034	59.35±0.98	0.624±0.015	0.595±0.011
NN methods	Acc [%]	AUC	*F*-score	Acc [%]	AUC	*F*-score
$DFFNN_{ngram}$ [10]	86.19±5.71	0.931±0.034	0.894±0.044	81.98±0.79	0.881±0.005	0.813±0.010
SCNN [43]	87.81±3.94	0.925±0.036	0.906±0.031	80.62±0.62	0.863±0.007	0.798±0.007
$DFFNN_{skipgram}$ [10]	64.16±0.89	0.646±0.066	0.781±0.006	78.85±1.00	0.860±0.009	0.777±0.011
CNN_{cbow} [61]	77.96±7.68	0.818±0.100	0.839±0.048	79.64±0.75	0.867±0.008	0.786±0.009
DFFNN (this study)	86.21±3.93	0.932±0.030	0.893±0.028	**82.80±0.50**	**0.893±0.006**	**0.825±0.005**
CNN (this study)	**88.35±3.29**	**0.946±0.025**	**0.910±0.026**	81.30±0.72	0.879±0.008	0.806±0.009

Note: the best results are in bold, [a] obtained for $C=2^3$ and $C=2^5$ for the doctor and Amazon dataset, respectively

To compare the accuracies statistically, a nonparametric Friedman test [22] was performed across the four datasets. This test is based on ranking the methods according to the Friedman statistic. The Friedman *p*-value indicates the

significant differences between the tested fake detection methods. Among the methods, the CNN and DFFNN models ranked first and second, respectively. To further compare the results against the best performer (CNN used as a control method), the Holm–Bonferroni post-hoc procedure [22] was employed to adjust the significance level. Table 5 shows that all the baseline methods and $DFFNN_{skipgram}$ were significantly outperformed by the proposed CNN model, whereas $DFFNN_{ngram}$, SCNN, CNN_{cbow} and DFFNN performed statistically similarly at p=0.05 in terms of accuracy.

To compare the computational time of the proposed models, we adopted the approach used in previous studies [9, 17] and used testing times to demonstrate real-time capacity. The results in Table 6 show that the proposed models were less time efficient than the other NN models. However, the capacity of the proposed models can be considered to be sufficient for online detection systems because approximately 3,000 reviews can be categorized per second. For example, the average testing times for CNNs were 2,446 reviews/sec, 3,333 reviews/sec, 3,610 reviews/sec and 4,631 reviews/sec for the hotel, restaurant, doctor and Amazon datasets, respectively, indicating acceptable throughput of the proposed fake detection system irrespective of data size and review domain.

Table 5: Results of Friedman nonparametric test

Method	Aver. ranking	*p*-value (vs. CNN)
SVM	7.00	0.050*
NB	7.50	0.031*
Bagging	8.25	0.014*
k-NN	10.5	0.001*
AdaBoost	9.75	0.002*
RF	9.25	0.004*
$DFFNN_{ngram}$ [10]	2.88	0.731
SCNN [43]	3.00	0.695
$DFFNN_{skipgram}$ [10]	9.50	0.003*
CNN_{cbow} [61]	6.25	0.096
DFFNN (this study)	2.13	0.961
CNN (this study)	2.00	–
Friedman *p*-value	0.0003*	

* statistically significant difference at p=0.05

Table 6: Testing time for comparing the fake review detection methods

	Hotel dataset	Restaurant dataset	Doctor dataset	Amazon dataset
Method	Testing time [s]	Testing time [s]	Testing time [s]	Testing time [s]
SVM	0.001±0.000	0.000±0.000	0.000±0.001	0.291±0.026
NB	0.106±0.002	0.040±0.001	0.003±0.000	3.612±0.217
Bagging	0.001±0.000	0.000±0.000	0.000±0.000	0.034±0.007
k-NN	1.340±0.011	0.100±0.002	0.016±0.006	296.241±2.022

AdaBoost	0.000±0.000	0.000±0.000	0.000±0.000	0.016±0.000
RF	0.011±0.001	0.002±0.001	0.001±0.001	0.059±0.014
$DFFNN_{ngram}$ [10]	0.211±0.011	0.064±0.005	0.087±0.004	15.208±6.230
SCNN [43]	0.953±0.017	0.081±0.007	0.046±0.001	2.551±0.006
$DFFNN_{skipgram}$ [10]	0.015±0.001	0.005±0.008	0.006±0.001	0.859±0.001
CNN_{cbow} [61]	0.084±0.001	0.023±0.008	0.032±0.001	0.877±0.007
DFFNN (this study)	0.271±0.005	0.101±0.004	0.127±0.002	7.894±0.561
CNN (this study)	0.327±0.007	0.120±0.002	0.154±0.027	4.535±0.016

Note: The experiments were performed using Intel i5-8400 2.80 GHz with six cores/threads and 16 GB RAM in the Weka 3.8.3 x64 program environment on a Windows 10 operating system. Deep NNs were implemented in the Deeplearning4j Java library.

Finally, Table 7 shows the results for the ARR measure, indicating the ratio between accuracy and testing time. To calculate the average ranking for each value of *Imp*, the geometric mean was calculated across all the datasets and the arithmetic mean ARR was then obtained across the compared methods, as suggested in [13]. As expected, the rankings of fast detection methods such as SVM and AdaBoost improved when testing time was considered to be the dominant criterion (*Imp* = 10%). On the contrary, the proposed models performed best when accuracy was considered to be more important (*Imp* = 0.1%) or both criteria were equally important (*Imp* = 1%), a scenario close to the real-world situation.

Table 7: Average rankings based on the ARR measure for the three values of Imp

	Imp = 0.1%[a]		*Imp* = 1%		*Imp* = 10%[b]	
Method	ARR	Aver. ranking	ARR	Aver. ranking	ARR	Aver. ranking
SVM	1.023	5	1.036	5	1.214	1
NB	0.948	8	0.945	9	0.926	11
Bagging	0.970	7	0.985	7	1.182	3
k-NN	0.919	10	0.908	12	0.816	12
AdaBoost	0.952	5	0.970	8	1.212	2
RF	0.915	11	0.921	11	1.005	5
$DFFNN_{ngram}$ [10]	1.090	3	1.080	3	1.001	8
SCNN [43]	1.088	4	1.079	4	1.006	4
$DFFNN_{skipgram}$ [10]	0.939	9	0.940	10	0.965	10
CNN_{cbow} [61]	1.019	6	1.016	6	0.993	9
DFFNN (this study)	1.097	1	1.087	1	1.003	6
CNN (this study)	1.096	2	1.086	2	1.002	7

Note: [a] accuracy is the dominant criterion, [b] testing time is the dominant criterion.

6 Conclusion

In this study, we proposed two deep NN models for detecting consumer fake reviews using an integrated framework of *n*-gram, Skip-Gram and emotion models. Using such complex high-dimensional models seems to be necessary to outperform existing approaches. The experimental results performed on four real-life fake review datasets demonstrate the effectiveness of the proposed models. Notably, the proposed models outperformed existing baseline approaches and state-of-the-art fake review detection methods in terms of accuracy, AUC and *F*-score. The experimental results also showed that the proposed integrated models were the most effective for larger datasets with combined polarity, implying their potential applications in real-life scenarios. To achieve high accuracy in such a setting, we leveraged pre-trained word embeddings and different emotion representations. Importantly, the proposed detection systems were also effective in terms of time complexity and detection time, as indicated by the ARR multicriteria measure.

A major limitation of the proposed model is that reviewer- and product-based features were not fully used. Compared with the multi-modal embedding representation proposed in [47], rich behaviour features were neglected, such as the ratio of reviewers' positive/negative reviews and the rating distribution of a reviewer's (product's) reviews. It is therefore recommended that further research should combine the proposed models with graph-based approaches using review metadata. For future works, we also plan to use the advantages of both the DFFNN and the CNN models and develop a hybrid deep NN structure similar to that of the "Network in Network" [45]. Such a hybrid model could further improve the generalization ability of the CNN model. Another disadvantage of the proposed model is that in contrast to the CNN model in [43], sentence weights were ignored due to their domain-specific nature. However, they can also be incorporated into the CNN model to consider the importance of different sentences. The results of this study suggest that the proposed deep NN models might have great potential application in related challenging text classification tasks such as fake news detection and opinion spam detection. Word embeddings can be easily pre-trained on other large corpuses for alternative domain applications. To obtain the complete sentence representation, the emotion indicators used in the proposed model can also be applied to other domains because they are based on general purpose lexicons.

Acknowledgments. This article was supported by the scientific research project of the Czech Sciences Foundation Grant No: 19-15498S and by the Operational Program: Research and Innovation project "Fake news on the Internet - identification, content analysis, emotions", co-funded by the European Regional Development Fund.

Conflict of Interest Statement: The authors declare that they have no conflict of interest.

References

1. Ahmed K, El Tazi N, Hossny AH (2015) Sentiment analysis over social networks: An overview. In: 2015 IEEE Int. Conf. on Systems, Man, and Cybernetics, IEEE, pp 2174–2179. doi: 10.1109/SMC.2015.380.
2. Ahmed H, Traore I, Saad S (2018) Detecting opinion spams and fake news using text classification. Security and Privacy 1(1):e9. doi: 10.1002/spy2.9.
3. Asghar MZ, Ullah A, Ahmad S, Khan A (2019) Opinion spam detection framework using hybrid classification scheme. Soft Computing, 17–24. doi: 10.1007/s00500-019-04107-y.
4. Baccianella S, Esuli A, Sebastiani F (2010) Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. Lrec 10:2200–2204.
5. Barbado R, Araque O, Iglesias CA (2019) A framework for fake review detection in online consumer electronics retailers. Information Processing & Management 56(4):1234–1244. doi: 10.1016/j.indmarman.2019.08.003.
6. Barushka A, Hajek P (2016) Spam filtering using regularized neural networks with rectified linear units. In: Adorni G, Cagnoni S, Gori M, Maratea M (eds) Conf. of the Italian Association for Artificial Intelligence. Lecture Notes in Computer Science 10037. Springer, Cham, pp 65–75. doi: 10.1007/978-3-319-49130-1_6.
7. Barushka A, Hajek P (2018) Spam filtering in social networks using regularized deep neural networks with ensemble learning. In: Iliadis L, Maglogiannis I, Plagianakos V (eds) Artificial Intelligence Applications and Innovations. AIAI 2018. IFIP Advances in Information and Communication Technology 519. Springer, Cham, pp 38–49. doi: 10.1007/978-3-319-92007-8_4.
8. Barushka A, Hajek P (2018) Spam filtering using integrated distribution-based balancing approach and regularized deep neural networks. Applied Intelligence 48(10):3538–3556. doi: 10.1007/s10489-018-1161-y.

9. Barushka A, Hajek P (2019) Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. Neural Computing and Applications, 1–19. doi: 10.1007/s00521-019-04331-5.
10. Barushka A, Hajek P (2019) Review spam detection using word embeddings and deep neural networks. In: MacIntyre J, Maglogiannis I, Iliadis L, Pimenidis E (eds) Artificial Intelligence Applications and Innovations. AIAI 2019. IFIP Advances in Information and Communication Technology 559. Springer, Cham, pp 340–350. doi: 10.1007/978-3-030-19823-7_28.
11. Bravo-Marquez F, Frank E, Mohammad SM, Pfahringer B (2016) Determining word-emotion associations from tweets by multi-label classification. In: 2016 IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI). IEEE, pp 536–539. doi: 10.1109/WI.2016.0091.
12. Bravo-Marquez F, Mendoza M, Poblete B (2014) Meta-level sentiment models for big social data analysis. Knowledge-Based Systems 69:86–99. doi: 10.1016/j.knosys.2014.05.016.
13. Brazdil PB, Soares C, Da Costa JP (2003) Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning 50(3):251–277. doi: 10.1023/A:102171390.
14. BrightLocal (2018) Local consumer review survey 2018. Available at: https://www.brightlocal.com/research/local-consumer-review-survey/.
15. Chandy R, Gu H (2012) Identifying spam in the iOS app store. In: Proc. of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality. ACM, pp 56–59. doi: 10.1145/ 2184305.2184317.
16. Chatzakou D, Vakali A (2015) Harvesting opinions and emotions from social media textual resources. IEEE Internet Computing 19(4):46–50. doi: 10.1109/MIC.2015.28.
17. Chen W, Yeo CK, Lau CT, Lee BS (2017) A study on real-time low-quality content detection on Twitter from the users' perspective. PLoS ONE 12(8):e0182487. doi: 10.1371/journal.pone.0182487.
18. Crawford M, Khoshgoftaar TM, Prusa JD, Richter AN, Al Najada H (2015) Survey of review spam detection using machine learning techniques. Journal of Big Data 2(1):1–23. doi: 10.1186/s40537-015-0029-9.
19. Elmurngi E, Gherbi A (2017) An empirical study on detecting fake reviews using machine learning techniques. In: 7th Int. Conf. on Innovative Computing Technology (INTECH). IEEE, pp 107–114. doi: 10.1109/INTECH.2017. 8102442.

20. Felbermayr A, Nanopoulos A (2016) The role of emotions for the perceived usefulness in online customer reviews. Journal of Interactive Marketing 36:60–76. doi: 10.1016/j.intmar.2016.05.004.
21. Floyd K, Freling R, Alhoqail S, Cho HY, Freling T (2014) How online product reviews affect retail sales: A meta-analysis. Journal of Retailing 90(2):217–232. doi: 10.1016/j.jretai.2014.04.004.
22. Garcia S, Fernandez A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Information Sciences 180(10):2044–2064. doi: 10.1016/j.ins.2009.12.010.
23. Garcia L (2018) Deception on Amazon – An NLP exploration, https://medium.com/@lievgarcia/deception-on-amazon-c1e30d977cfd, last accessed 2019/09/01.
24. Ghai R, Kumar S, Pandey AC (2019) Spam detection using rating and review processing method. In: Smart Innovations in Communication and Computational Sciences. Springer, Singapore, pp 189–198. doi: 10.1007/978-981-10-8971-8_18.
25. Hajek P (2018) Combining bag-of-words and sentiment features of annual reports to predict abnormal stock returns. Neural Computing and Applications 29(7):343–358. doi: 10.1007/s00521-017-3194-2.
26. Harris C (2012) Detecting deceptive opinion spam using human computation. In: Workshops at AAAI on Artificial Intelligence. AAAI, pp 87–93.
27. He R, McAuley J (2016) Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proc. of the 25th Int. Conf. on World Wide Web, pp 507–517. doi: 10.1145/2872427.2883037.
28. Heydari A, ali Tavakoli M, Salim N, Heydari Z (2015) Detection of review spam: A survey. Expert Systems with Applications 42(7):3634–3642. doi: 10.1016/j.eswa.2014. 12.029.
29. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proc. of the 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. ACM pp 168-177. doi: 10.1145/1014052.1014073.
30. Hussain N, Turab Mirza H, Rasool G, Hussain I, Kaleem, M (2019) Spam review detection techniques: A systematic literature review. Applied Sciences 9(5):987. doi: 10.3390/app9050987.
31. Ikeda K, Hattori G, Ono C, Asoh H, Higashino T (2013) Twitter user profiling based on text and community mining for market analysis. Knowledge-Based Systems 51:35–47. doi: 10.1016/j.knosys.2013.06.020.

32. Jain G, Sharma M, Agarwal B (2018) Spam detection on social media using semantic convolutional neural network. International Journal of Knowledge Discovery in Bioinformatics (IJKDB) 8(1):12–26. doi: 10.4018/IJKDB.2018010102.

33. Jain G, Sharma M, Agarwal B (2019) Spam detection in social media using convolutional and long short term memory neural network. Annals of Mathematics and Artificial Intelligence 85(1):21–44. doi: 10.1007/s10472-018-9612-z.

34. Jindal N, Liu B (2007) Analyzing and detecting review spam. In: 7th IEEE Int. Conf. on Data Mining. ICDM 2007, IEEE, pp 547–552. doi: 10.1109/ICDM.2007.68.

35. Kennedy S, Walsh N, Sloka K, McCarren A, Foster J (2019) Fact or factitious? Contextualized opinion spam detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. ACL, pp. 344–350. doi: 10.18653/v1/P19-2048.

36. Kiritchenko S, Zhu X, Mohammad SM (2014) Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research 50:723–762.

37. Lau RY, Liao SY, Kwok RCW, Xu K, Xia Y, Li Y (2011) Text mining and probabilistic language modeling for online review spam detecting. ACM Transactions on Management Information Systems 2(4):1–30. doi: 10.1145/2070710.2070716.

38. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International Conference on Machine Learning. JMLR 32, pp. 1188–1196.

39. Li H, Chen Z, Mukherjee A, Liu B, Shao J (2015) Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In: 9th Int. AAAI Conf. on Web and Social Media (ICWSM 2015). AAAI, pp 634–637.

40. Li H, Fei G, Wang S, Liu B, Shao W, Mukherjee A, Shao J (2017) Bimodal distribution and co-bursting in review spam detection. In: 26th Int. Conf. on World Wide Web. ACM, pp 1063–1072. doi: 10.1145/3038912.3052582.

41. Li F, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. In: Int. Joint Conf. on Artificial Intelligence (IJCAI 2011), pp. 2488–2493 (2011).

42. Li J, Ott M, Cardie C, Hovy E (2014) Towards a general rule for identifying deceptive opinion spam. In: Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics. ACL 1, pp 1566–1576. doi: 10.3115/v1/P14-1147.

43. Li L, Qin B, Ren W, Liu T (2017) Document representation and feature combination for deceptive spam review detection. Neurocomputing 254:33–41. doi: 10.1016/j.neucom.2016.10.080.

44. Lim EP, Nguyen VA, Jindal N, Liu B, Lauw HW (2010) Detecting product review spammers using rating behaviors. In: 19th ACM Int. Conf. on Information and Knowledge Management. ACM, pp 939–948. doi: 10.1145/1871437.1871557.

45. Lin M, Chen Q, Yan S (2014) Network in network. Int. Conf. on Learning Representations (ICLR). ICLR, pp 1–10.

46. Liu Y, Pang B (2018) A unified framework for detecting author spamicity by modeling review deviation. Expert Systems with Applications 112:148–155. doi 10.1016/j.eswa.2018.06.028.

47. Liu Y, Pang B, Wang X (2019) Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing 366:276–283. doi: 10.1016/j.neucom.2019. 08.013.

48. Madisetty S, Desarkar MS (2018) A neural network-based ensemble approach for spam detection in Twitter. IEEE Transactions on Computational Social Systems 5(4):973–984. doi: 10.1109/TCSS.2018.2878852.

49. Malik MSI, Hussain A (2017) Helpfulness of product reviews as a function of discrete positive and negative emotions. Computers in Human Behavior 73:290–302. doi: 10.1016/j.chb.2017.03.053.

50. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. NIPS 26, pp 3111–3119.

51. Mohammad SM, Turney PD (2013) Crowdsourcing a word–emotion association lexicon. Computational Intelligence 29(3):436–465. doi: 10.1111/j.1467-8640.2012.00460.x.

52. Mukherjee A, Venkataraman V, Liu B, Glance N (2013) What yelp fake review filter might be doing?. In: 7th Int. AAAI Conf. on Weblogs and Social Media. AAAI, pp 409–418.

53. Nielsen FÅ (2011) A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. Proc. of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, pp 93–98.

54. Ott M, Choi Y, Cardie C, Hancock JT (2011) Finding deceptive opinion spam by any stretch of the imagination. In: Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. ACL, pp 309–319.

55. Ott M, Cardie C, Hancock J (2012) Estimating the prevalence of deception in online review communities. In: 21st Int. Conf. on World Wide Web. ACM, pp 201–210. doi: 10.1145/2187836.2187864.

56. Ott M, Cardie C, Hancock JT (2013) Negative deceptive opinion spam. In: 2013 Conf. of the North American Chapter of the Assoc. for Computational Linguistics: Human Language Technologies. ACL, pp 497–501.

57. Pandey AC, Rajpoot DS (2019) Spam review detection using spiral cuckoo search clustering method. Evolutionary Intelligence 12(2):147–164. doi: 10.1007/s12065-019-00204-x.

58. Patel NA, Patel R (2018) A survey on fake review detection using machine learning techniques. In: 2018 4th Int. Conf. on Computing Communication and Automation (ICCCA). IEEE, pp 1–6. doi: IEEE.10.1109/ CCAA.2018.8777594.

59. Peng Q, Zhong M (2014) Detecting spam review through sentiment analysis. Journal of Software 9(8):2065–2072. doi: 10.4304/jsw.9.8.2065-2072.

60. Rayana S, Akoglu L (2015) Collective opinion spam detection: Bridging review networks and metadata. In: 21th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. ACM, pp 985–994. doi: 10.1145/2783258.2783370.

61. Ren Y, Ji D (2017) Neural networks for deceptive opinion spam detection: An empirical study. Information Sciences 385:213–224. doi: 10.1016/j.ins.2017.01.015.

62. Rout JK, Dalmia A, Choo KKR, Bakshi S, Jena SK (2017) Revisiting semi-supervised learning for online deceptive review detection. IEEE Access 5:1319–1327. doi: 10.1109/ACCESS.2017.2655032.

63. Rout JK, Dash AK, Ray NK (2018) A framework for fake review detection: Issues and challenges. In: 2018 Int. Conf. on Information Technology (ICIT). IEEE, pp 7–10. doi: 10.1109/ICIT.2018.00014.

64. Shojaee S, Murad MAA, Azman AB, Sharef NM, Nadali S (2013) Detecting deceptive reviews using lexical and syntactic features. In: 13th Int. Conf. on Intellient Systems Design and Applications. IEEE, pp 53–58. doi: 10.1109/ISDA. 2013.6920707.

65. Sun C, Du Q, Tian G. (2016) Exploiting product related review features for fake review detection. Mathematical Problems in Engineering 2016:1–7. doi: 10.1155/2016/4935792.
66. Tang X, Qian T, You Z (2019) Generating behavior features for cold-start spam review detection. In: Int. Conf. on Database Systems for Advanced Applications. Springer, Cham, pp 324–328. doi: 10.1007/978-3-030-18590-9_38.
67. The Times (2018) 'A third of TripAdvisor reviews are fake' as cheats buy five stars. The Times September 22 2018. Available at: https://www.thetimes.co.uk/article/hotel-and-caf-cheats-are-caught-trying-to-buy-tripadvisor-stars-027fbcwc8, last accessed 2019/01/22.
68. TripAdvisor Homepage, http://ir.tripadvisor.com/, last accessed 2019/01/21.
69. Vidanagama DU, Silva TP, Karunananda AS (2019) Deceptive consumer review detection: a survey. Artificial Intelligence Review, 1–30. doi: 10.1007/s10462-019-09697-5.
70. Wang G, Xie S, Liu B, Philip SY (2011) Review graph based online store review spammer detection. In: 11th Int. Conf. on Data mining (ICDM 2011). IEEE, pp 1242–1247. doi: 10.1109/ICDM.2011.124.
71. Xie S, Wang G, Lin S, Yu PS (2012) Review spam detection via temporal pattern discovery. In: 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. ACM, pp 823–831. doi: 10.1145/2339530.2339662.
72. Xue H, Wang Q, Luo B, Seo H, Li F (2019) Content-aware trust propagation toward online review spam detection. Journal of Data and Information Quality (JDIQ) 11(3):11. doi: 10.1145/3305258.
73. Ye J, Kumar S, Akoglu L (2016) Temporal opinion spam detection by multivariate indicative signals. In: 10th Int. AAAI Conf. on Web and Social Media (ICWSM 2016). AAAI, pp 743–746.
74. Yilmaz CM, Durahim AO (2018) SPR2EP: A semi-supervised spam review detection framework. In: 2018 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, pp 306–313. doi: 10.1109/ASONAM.2018.8508314.
75. Wang G, Li C, Wang W, Zhang Y, Shen D, Zhang X, Henao R, Carin L (2018) Joint embedding of words and labels for text classification. In: Proc. the 56th Annual Meeting of the Association for Computational Linguistics. ACL, pp 2321–2331. doi: 10.18653/v1/P18-1216.

76. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proc. of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. ACL, pp 347–354.

77. Zeng ZY, Lin JJ, Chen MS, Chen MH, Lan YQ, Liu JL (2019) A review structure based ensemble model for deceptive review spam. Information 10(7):243. doi: 10.3390/info10070243.