# Deceptive Reviews Detection Using Deep Learning Techniques

Nishant Jain(✉), Abhay Kumar(✉), Shekhar Singh(✉), Chirag Singh(✉), and Suraj Tripathi(✉)

Samsung R&D Institute India, Bangalore, India
{nishant.jain, abhayl.kumar, s.singh02, c.singh, suraj.tri}@samsung.com

**Abstract.** With the increasing influence of online reviews in shaping customer decision-making and purchasing behavior, many unscrupulous businesses have a vested interest in generating and posting deceptive reviews. Deceptive reviews are fictitious reviews written deliberately to sound authentic and deceive the consumers. Traditional deceptive reviews detection methods are based on various handcrafted features, including linguistic and psychological, which characterize the deceptive reviews. However, the proposed deep learning methods have better self-adaptability to extract the desired features implicitly and outperform all traditional methods. We have proposed multiple Deep Neural Network (DNN) based approaches for deceptive reviews detection and have compared the performances of these models on multiple benchmark datasets. Additionally, we have identified a common problem of handling the variable lengths of these reviews. We have proposed two different methods – Multi-Instance Learning and Hierarchical architecture to handle the variable length review texts. Experimental results on multiple benchmark datasets of deceptive reviews have outperformed existing state-of-the-art. We evaluated the performance of the proposed method on other review-related task-like review sentiment detection as well and achieved state-of-the-art accuracies on two benchmark datasets for the same.

**Keywords:** Deceptive reviews · Fake reviews · Deep learning · Convolutional neural network · Recurrent neural network · Word embedding

## 1 Introduction

In recent years, there has been a dramatic increase in the number of online user-generated reviews for a plethora of products and services across multiple websites. These reviews contain the subjective opinion of the users along with various detailed information. We rely a lot on these user-reviews before making up our mind, like which restaurant to go, what to buy, which hotel to stay in, and so on. Given the increased influence of these reviews in shaping customer's decision making, there is an incentive and opportunities for unscrupulous business to generate and post fake

N. Jain and A. Kumar—equal contribution. S. Singh, C. Singh and S. Tripathi—equal contribution.


E. Métais et al. (Eds.): NLDB 2019, LNCS 11608, pp. 79–91, 2019.
https://doi.org/10.1007/978-3-030-23281-8_7

reviews, either in favor of themselves or in disapproval of competition rivals. Deceptive reviews are deliberately written to sound authentic and help businesses to gain financial advantage and enhance their reputation. In addition, online reviews are of varied writing styles, linguistic types, content and review lengths, making it difficult for human readers to identify themselves as well.

With ever-increasing instances of deceptive reviews, there has been a series of research works to identify deceptive/fake reviews using different linguistic and psychological cues. In a marketing research study by Spiegel[1], it has been shown that nearly 95% of shoppers make a purchase after reading online reviews and that the product with at least five reviews has 270% greater likelihood to be purchased than the products with no reviews. This shows the necessity of robust deception detection methods to maintain the reliability and facticity of online reviews. Recently, this has captured the attention of both businesses and research community; giving rise to state-of-the-art results.

## 2 Related Work

Spam detection has been historically researched extensively in the contexts of e-mails [1] and web-texts [2]. In recent years, researchers have proposed various approaches for deceptive or manipulative reviews detection. Jindal et al. [3] proposed a supervised classifier (Logistic Regression) using features based on review content, reviewer profile, and the product descriptions. Yoo et al. [4] presented the comparative study of language structure of truthful and deceptive reviews using deception theory and demonstrated the difficulty of detecting deceptive reviews based on the structural properties, i.e. lexical complexity. Ott et al. (2011) [5] employed Turkers to write deceptive reviews and created a benchmark dataset of 800 reviews (400 gold-standard deceptive reviews and 400 truthful reviews) to be used in subsequent works. They modeled it as n-gram based text categorization task and proposed a Support Vector Machine (SVM) classifier exploiting the computational linguistics and psychological approaches for detecting deceptive reviews. They have also framed it as a genre classification task, exploiting the writing style difference between informative and imaginative reviews for truthful and deceptive reviews respectively. Additionally, they have assessed the human performance for the task; the average accuracy of three human judges were meager 57.3% as compared to 89.9% accuracy of their proposed classifier. Feng et al. [6] investigated the syntactic stylometry approach and achieved better performance by using syntactic features from context-free-grammar parse trees. Most of these works were focused on extracting the richer textual features to improve deception detection performance. However, the difficulty of creating human-labeled data and the inability of hand-crafted features to capture non-local semantic information over a discourse solicited various alternative approaches, like semi-supervised learning approach, approaches exploiting the user behavioral aspects, etc.

[1] https://spiegel.medill.northwestern.edu/online-reviews/.

Feng et al. [7] used aspect based profile compatibility measure to compare the test review with the product profile, built from a separate collection of reviews for the same product. Mukherjee et al. [8] modeled the spamicity of an author using various observed reviewer's behavior to identify spammer. Apart from the textual features, many other works also have been focused on the behavioral aspects (like extreme ratings, too many reviews in short time, duplicate content or ratings, etc.) of the spammers. Ren et al. [9] proposed and Fusilier et al. [20] improved the semi-supervised learning method to detect deceptive reviews.

In recent years, Deep Neural Network (DNN) models have been used to learn better semantic representations for improved performance in various NLP tasks. Kim [10] introduced Convolutional Neural Network (CNN) model for text classification to capture the frame-based semantic features. Ren et al. [11] explore a neural network model to learn document-level representation for detecting deceptive reviews. They make use of gated recurrent neural network model with attention mechanism for detecting deceptive opinion; by capturing non-local discourse information over sentence vectors. Zhao et al. [12] use a convolution neural network model by embedding the word order characteristics in its convolution and pooling layer; this makes the model more efficient in detecting deceptive opinions.

## 3 Datasets

We evaluate our architectures quantitatively on three different benchmark datasets for deceptive reviews detection (Sects. 3.1 to 3.3). Additionally, we evaluated our proposed architecture for addressing variable length text sequences on another related task, i.e. Review Sentiment Detection. We evaluate our proposed model on two additional datasets (Sects. 3.4 and 3.5) for review sentiment detection to show the scalability of the proposed network for various text classification task. The statistics of these datasets is summarized in Table 1.

### 3.1 Deceptive Opinion Spam Corpus v1.4 (DOSC)

Deceptive opinion dataset [5] consists of real and fake reviews about 20 separate hotels in Chicago. It contains 400 real and 400 fake reviews of both positive and negative sentiments respectively. The truthful reviews have been collected from online websites like TripAdvisor[2], Expedia[3] etc., while the deceptive opinions have been collected using Amazon's Mechanical Turk. It also provides a predetermined five folds for 5-fold cross-validation.

[2] https://www.tripadvisor.com.

[3] https://www.expedia.com/.

### 3.2 Four-City Dataset

Four-city Dataset [13] consists of 40 real and 40 fake reviews for each of the eight hotels in four different cities. The real reviews were chosen using random sampling on positive 5-star reviews. Amazon's Mechanical Turk was used to write fake reviews to get gold-standard deception dataset.

### 3.3 YelpZip Dataset

YelpZip dataset [14] consists of real world reviews of restaurants and hotels sampled from yelp along with near ground truth as provided by the Yelp review filter. YelpZip consists of reviews from 5044 hotels by 260,277 reviewers from various New York State zip codes. There are 608598 total reviews with 528141 true and 80457 deceptive review dataset. Due to high data-imbalance, Fontanarava et al. [15] created a balanced dataset by under-sampling the truthful reviews. We follow the same setting as well.

### 3.4 Large Movie Review Dataset

Large Movie Review Dataset (LMRD) [16] is a sentiment classification benchmark dataset that contains 50,000 reviews from IMDB, with no more than 30 reviews per film. The dataset provides a train-test split with both training and testing split containing 25,000 reviews respectively. Both splits are further evenly divided into positive and negative reviews with 12,500 reviews in each category.

### 3.5 Drug Review Dataset

Drug Review Dataset (DRD) [17] is a review sentiment dataset with 215063 reviews from drugs.com website. The reviews in this dataset have three different polarities – positive, negative and neutral. The dataset provides a 75%–25% train-test split using stratified random sampling.

**Table 1.** Dataset statistics

| Dataset name | Total number of reviews | Word-length of review text | | | |
|---|---|---|---|---|---|
| | | Minimum | Maximum | Mean | Standard deviation |
| YelpZip | 608598 | 1 | 5213 | 115 | 106 |
| DOSC | 1600 | 26 | 784 | 149 | 87 |
| Four-city | 640 | 4 | 413 | 138 | 47 |
| DRD | 215063 | 1 | 1857 | 86 | 46 |
| LMRD | 50000 | 4 | 2470 | 234 | 173 |

### 3.6 Dataset Statistics and Visualization

We have shown (in Fig. 1) the word-cloud diagrams to visualize the word frequencies in truthful and deceptive reviews. Both the word-clouds have occurrences of very similar words. We found that the frequencies of the top hundred words per review in both truthful and deceptive reviews are similar. This makes it very difficult for traditional methods like *tf-idf* to identify the deceptive reviews correctly. This calls for deep learning based approaches to learn the semantic and syntactic differences between the truthful and deceptive reviews. We have also shown (in Fig. 2) the frequency distribution of data with respect to the length for two different datasets.

**Fig. 1.** Word-cloud of frequently used words in truthful and deceptive reviews

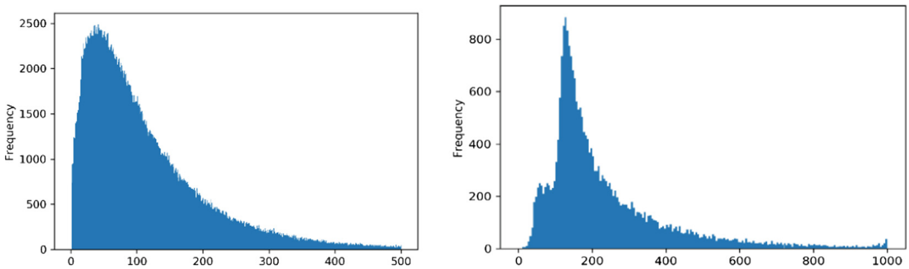


**Fig. 2.** Histogram plot of sequence length of reviews (for Yelp & LMRD datasets respectively)

## 4 Neural Network Models

For both Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), due to computational limitations, we decided to trim the input sequence to 150 words (average word-length for all the datasets derived from the statistic shown in Table 1) since this led to reasonable convergence time and memory footprint.

Also, inputs (reviews) are represented using word embedding as discussed below.

### 4.1 Word Embedding

Word embedding gained popularity through its use in various NLP tasks say, language modelling, text classification and sentiment analysis in recent past. Word embedding is a distributed representations of a word in an n-dimensional space using high-dimensional vectors (say, 200 to 500 dimensions).

Mikolov et al. [18] proposed an unsupervised architecture to learn the distributed representation of words in a corpus. He gave two different architecture namely continuous bag of words (CBOW) and skip-gram. CBOW predicts the current word based on surrounding words while skip-gram predicts the surround words based on the given word. Mikolov et al. [18] also provided pre-trained word embedding based on google news corpus.

We experimented with both pre-trained and randomly initialized word embedding in our experiment and found pre-trained embedding to be more accurate. We also experimented with training our own word vector and have discussed its effects in the discussions section below.

### 4.2 Convolutional Neural Network

CNN uses convolutional and pooling to extract spatial features based on the locality of reference in images. In recent studies, CNNs have been extended to NLP tasks as well. Kim [10] showed that CNN can be used effectively for the text classification task and gives promising results. In our model, we use three parallel convolutional layers with 100 filters each. The kernel shape for the three convolutional blocks are-$f_i^h \times f_i^w$, where, $f_i^w = dimension\ of\ word\ embedding\ vector$ and $f_i^h = 3, 5, \&7$. The feature maps generated by all three convolutional blocks are max-pooled, concatenated and fed to the hidden fully connected (FC) layer of 1024 and 256 neurons sequentially. The output of hidden FC is fed to the output softmax layer to give the class probabilities. Dropout regularization is also used between concatenation layer and first FC layer to counter the effect of overfitting (Fig. 3).

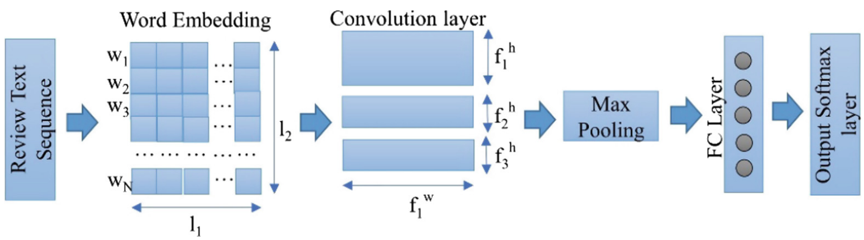


**Fig. 3.** Architecture diagram of convolutional neural network

### 4.3 Recurrent Neural Network

CNN is a great tool to extract features from a local region but less effective while learning long term dependencies. To overcome this obstacle, RNNs are used which are capable of learning features from long term dependencies through their recurrent structure. But the vanilla RNN in practice suffers from problems of short-term memory. They are not able to retain information over longer sentences due to vanishing gradient problem. To handle this issue, we use Gated Recurrent Unit (GRU) [19] in our experiments, which can overcome this problem by regulating the flow of information through them.

The architecture consists of a single GRU of 1024 units along with attention module. The attention module assigns a value between zero and one to each word; depicting its importance or relevance to the context vector. The output of GRU weighted by attention values is fed to the FC layer, which outputs the final class probability distribution. The GRU unit can be defined as below:-

$$z_t = \sigma_g \left( W_z x_t + U_z h_{t-1} + b_z \right) \tag{1}$$

$$r_t = \sigma_h \left( W_r x_t + U_r h_{t-1} + b_r \right) \tag{2}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \sigma_h (W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \tag{3}$$

where $x_t$ is input vector and $h_t$ is output vector at time t. $z_t$ is the update gate vector and $r_t$ is the reset gate vector. $W, U$ and $b$ are learnable matrices and vectors. $\sigma_g$ and $\sigma_h$ represents activation functions and $\odot$ is Hadamard operator and $h_0 = 0$.

## 5 Proposed Methods for Handling Variable Length Reviews

We observe a large variation in the sequence length of review texts, and therefore we need to decide the maximum sequence length of the input review texts. Usually, we set a maximum sequence length to get optimal computation cost and accuracies. If the max-sequence length is small, then some part of the text is trimmed and not exposed to the network. This affects the accuracies worst in those cases, where the user summarize their opinion in the last few sentences. Selecting larger sequence length leads to high computation cost and slow convergence. Having observed this common problem across various text classification task especially review-text related tasks, we proposed two different architectures to address the problem. The proposed models are described in the following sub-sections.

### 5.1 Hierarchical Model Architecture

We propose a hierarchical model architecture with CNN followed by GRU to take variable length text sequences as input. The input review text is divided into multiple instances, each having twenty words. The total number of instances is a variable depending on the actual word length of the given review. Each instance is fed to a CNN network to extract localized regional features from the words in the instances. These features from CNN are able to capture lexical n-gram characteristics by using convolutional kernels of different shapes. We use three different convolution blocks with dimension $f_i^h \times f_i^w$, where $f_i^w = dimension\ of\ word\ embedding\ vector$, and $f_i^h = 3, 5, \&7$. Different convolutional kernels are capable of capturing n-gram semantics of varying granularities, i.e., tri-gram, five-gram, and seven-gram. The convolutional layer features are fed to max-pooling layer and the output is flattened. The flattened output from the CNN network acts as instance representation. The CNN block is the same as the one described in Sect. 4.2.

At the next stage, different instance representations are passed to a GRU network. The GRU model is good at capturing long-range discourse structures among the instance representations. The output representation of the GRU is fed to the FC layer and subsequently to the softmax layer to predict the class label probabilities. The detailed architecture of the proposed hierarchical CNN-GRU network is shown in Fig. 4.

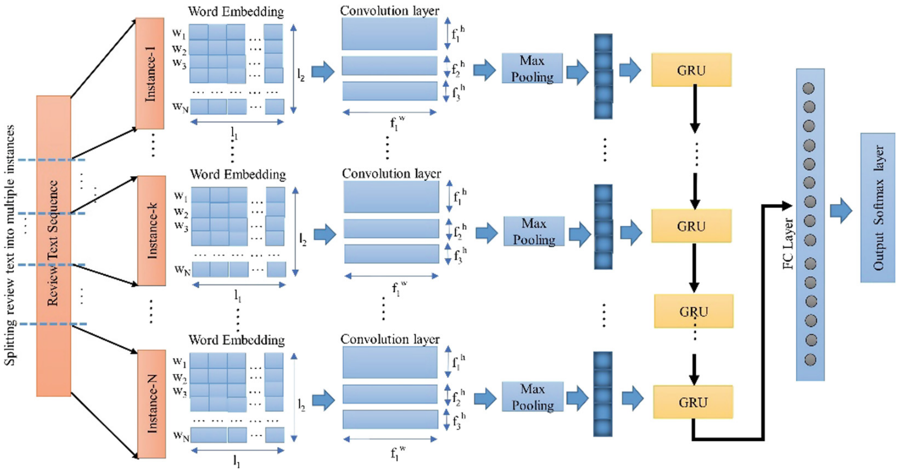


**Fig. 4.** Schematic diagram of hierarchical CNN-GRU model

## 5.2 Multi-Instance Learning (MIL)

We also propose a simplistic way to handle variable length review text using any deep learning architectures. In the multi-instance paradigm, we split the input text sequence into multiple instances of a fixed length (shown in Fig. 5). These different splits are fed as different instances to the model and act as different training examples with the same labels. We discard the last instance if its word-length is less than fifteen. During test time, we evaluate the class probabilities for all instances and assign the label by taking max-vote of predictions of all instances.

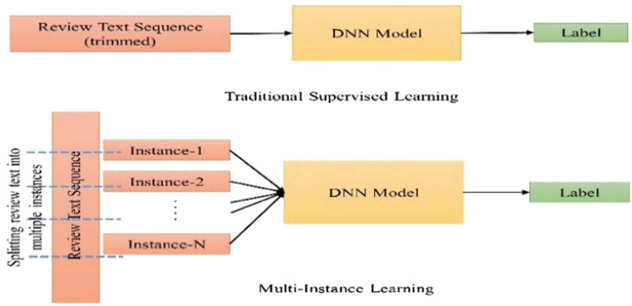


**Fig. 5.** Schematic diagram of traditional and multi-instance learning.

# 6 Evaluations and Discussions

We have performed a series of experiments for the deceptive review detection and the experimental results are presented and discussed in the following sub-sections. In case of no predefined train-test splits available for the dataset, we have split the data using stratified k-fold method and presented 5-fold cross validation scores.

## 6.1 Evaluation of Various DNN Based Models for Deceptive Review Detection

We have experimented with different model architectures for deceptive review detection. We have presented the 5-fold cross validation in Table 2. Baseline model accuracy for each dataset is also shown in the table for comparative study. Most literature

**Table 2.** Experimental results for various deceptive review datasets.

| Dataset | Models | Accuracy (%) | Precision | Recall |
|---|---|---|---|---|
| Deceptive opinion spam corpus v1.4 | Baseline [5] | 86.5 | 0.86 | 0.87 |
| | CNN | 89.6 | 0.89 | 0.89 |
| | GRU | 90.3 | 0.91 | 0.90 |
| | **MIL** | **90.1** | **0.90** | **0.90** |
| | **CNN-GRU** | **91.9** | **0.92** | **0.91** |
| Four-city dataset | Baseline [13] | 80.1 | 0.79 | 0.82 |
| | CNN | 82.4 | 0.82 | 0.81 |
| | GRU | 82.9 | 0.83 | 0.83 |
| | **MIL** | **82.8** | **0.83** | **0.83** |
| | **CNN-GRU** | **84.7** | **0.85** | **0.85** |
| YelpZip dataset | Baseline [15] | 54.2 | 0.63 | 0.48 |
| | CNN | 63.8 | 0.59 | 0.61 |
| | GRU | 64.2 | 0.63 | 0.62 |
| | **MIL** | **64.6** | **0.60** | **0.62** |
| | **CNN-GRU** | **66.4** | **0.67** | **0.65** |

on deceptive review detection covers traditional approaches by exploiting hand-crafted features- linguistic or psychological. DNN based models have outperformed traditional methods by adaptively learning the best possible contextual features, responsible to distinguish the truthful and deceptive reviews. For all three datasets, CNN and GRU models have achieved better accuracies, with GRU using attention mechanism being marginally better than CNN. GRU model is capable of capturing the long range dependencies inherent in the review text and giving adequate attention to the words aligned with the review context. Whereas, CNN fails to capture context-dependent semantic relationships in the long texts. For all three datasets, our models have outperformed state-of-the-art result by atleast 2.8%.

### 6.2 Evaluation of Proposed Models for Handling Variable Length Reviews

As evident from Table 1, the review texts are varied in its length and follow a long-tail distribution as shown in Fig. 2. Although majority of the reviews are less than roughly hundred words, there are many reviews with much larger word-length. Usually, the decision to trim the long review text assumes that the underlying semantic and syntactic features, responsible for distinguishing truthful and deceptive reviews are present throughout the review text. But, this assumption ignores the human tendency or standard writing structure to conclude important factors at the end of review texts. Owing to this fact, we don't want to discard any valuable part of the review text and expose the complete review during the model training process. Our claims are verified by the increased performances of both our proposed models as compared to the vanilla CNN and GRU models. In Table 2, we have made a comparative study of two proposed approaches for handling variable length deceptive reviews. Hierarchical CNN-GRU model outperforms CNN or GRU models by at least 1.6% for all three datasets.

### 6.3 Evaluation on Proposed Models for Handling Variable Length Reviews on Another Task (Review Sentiment Detection)

To show the effectiveness and scalability of our proposed models in handling variable length text sequences, we evaluated them on two benchmark review sentiment detection task as well. In Table 3, we have made a comparative study of proposed

**Table 3.** Accuracy for the proposed models for review sentiment detection datasets

| Dataset | Models | Accuracy (%) | Precision | Recall |
|---|---|---|---|---|
| **LMRD** | CNN | 86.5 | 0.87 | 0.86 |
| | GRU | 86.8 | 0.87 | 0.87 |
| | **MIL** | **87.1** | **0.87** | **0.87** |
| | **CNN-GRU** | **88.9** | **0.88** | **0.89** |
| **DRD** | CNN | 76.8 | 0.77 | 0.77 |
| | GRU | 76.3 | 0.76 | 0.76 |
| | **MIL** | **78.2** | **0.78** | **0.78** |
| | **CNN-GRU** | **83.8** | **0.84** | **0.83** |

approaches and standard DNN models. Hierarchical CNN-GRU model outperforms standard CNN/GRU models by 2.1% and 7% on LMRD and DRD datasets respectively. Improved performance of the proposed models on two different tasks and 5 different datasets illustrates the importance of considering the entire review to get the complete context and not missing out any important and distinguishing aspects.

### 6.4 Discussions

**Effect of Different Lengths of Review Texts:** Review texts vary a lot in its word length, ranging from 1 to 5213 in our datasets. In the vanilla CNN and GRU architectures, we need to decide the maximum sequence length of the input review text, and the text is either trimmed or zero-padded accordingly. However, by restricting the review text to a smaller fixed length, the models are not exposed to the complete review and hence perform poorly in learning the overall context of the reviews. In addition, adopting a larger maximum sequence length increases both the number of learnable parameters and computational cost. Our proposed models take a different number of instances depending on the length of an input review text and learns the complete context of the review. The better performance of our proposed models; confirms the hypothesis that discriminative semantic and syntactic features are not evenly distributed throughout the review texts and could be even present in the concluding sentences.

**Effect of Pre-trained Word Embeddings:** We used word2vec models to get word-embedding vector. We experimented with both pre-trained word2vec on Google News corpus and pre-trained on review dataset (combined corpus of all dataset mentioned in Sect. 3). There is a marginal improvement in accuracies by using pre-trained word2vec embedding on review datasets. Empirically, we find that 300-dimensional embedding performs better than 150-dimensional one.

## 7 Conclusions

In the paper, we have experimented with various deep learning based models for identifying deceptive reviews. We presented a comparative study of the experimental results of the various models on four different benchmark datasets. Additionally, we have identified a common problem across all datasets, i.e., variable length of the reviews. In other text classification task, we usually trim the long text sequences and zero-pad the short ones. However, we lose a significant part of the review's semantic information. We have proposed two different approaches to handle the high variance of review textual lengths. Multi-Instance Learning approach is based on feeding different instances of the same training example to the same model. Hierarchical CNN-GRU model is based on extracting n-gram like semantic features using Convolutional Neural Network (CNN) and learning semantic dependencies among the extracted features from CNN modules. Both these models are capable of handling very long reviews texts and are better at deception detection. We have demonstrated that the proposed MIL and Hierarchical CNN-GRU models outperform the classical CNN and RNN models on all four benchmark datasets. For the future work, we will consider adding metadata of the reviews in our proposed models to make it more robust and accurate. We will also study and analyse the effect of our models on different NLU tasks; involving long and variable length features.

## References


1. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: Proceedings of the 13th International Conference on Very Large Data Bases, vol. 30, pp. 576–587 (2004)
2. Ntoulas, A., Najork, M., Manasse, M., Fetterly, D.: Detecting spam web pages through content analysis. In: Proceedings of the 15th International Conference on World Wide Web, pp. 83–92. May 2006 (2016)
3. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 219–230, February 2008
4. Yoo, K.H., Gretzel, U.: Comparison of deceptive and truthful travel reviews. Information and Communication Technologies in Tourism. Springer, Vienna (2009)
5. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 309–319 (2011)
6. Feng, S., Banerjee, R., Choi, Y.: Syntactic stylometry for deception detection. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Paper, vol. 2, pp. 171–175, July 2012
7. Feng, V.W., Hirst, G.: Detecting deceptive opinions with profile compatibility. In: Proceedings of the Sixth IJCNLP, pp. 338–346 (2013)
8. Mukherjee, A., et al.: Spotting opinion spammers using behavioral footprints. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 632–640 (2013)
9. Ren, Y., Ji, D., Zhang, H.: Positive unlabeled learning for deceptive reviews detection. In EMNLP, pp. 488–498 (2014)
10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv: 1408.5882(2014)
11. Ren, Y., Ji, D.: Neural networks for deceptive opinion spam detection: an empirical study. Inf. Sci. **385**, 213–224 (2017)
12. Zhao, S., Xu, Z., Liu, L., Guo, M., Yun, J.: Towards accurate deceptive opinions detection based on word order-preserving CNN. Math. Probl. Eng., **2018** (2018)
13. Li, J., Ott, M., Cardie, C.: Identifying manipulated offerings on review portals. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1933–1942 (2013)
14. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: Proceedings of the 21 thacmsigkdd International Conference on Knowledge Discovery and Data Mining, pp. 985–994, August 2015
15. Fontanarava, J., Pasi, G., Viviani, M.: Feature analysis for fake review detection through supervised classification. In: IEEE International Conference on Data Science and Advanced Analytics, pp. 658–666, October 2017
16. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A. Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150, June 2011

17. Gräßer, F., Kallumadi, S., Malberg, H., Zaunseder, S.: Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In: Proceedings of the 2018 International Conference on Digital Health, pp. 121–125, April 2018
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781(2013)
19. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555(2014)
20. Fusilier, D.H., Montes-y-Gómez, M., Rosso, P., Cabrera, R.G.: Detecting positive and negative deceptive opinions using PU-learning. Inf. Process. Manag. **51**, 433–443 (2015)