Anomaly Detection and Opinion Mining on Textual Reviews with Deep Learning

1st Nushrat Humaira
School of Computing
Clemson University
Clemson, USA
nhumair@g.clemson.edu

2nd Dheeraj Chandra
School of Computing
Clemson University
Clemson, USA
drallab@g.clemson.edu

3rd Srinija Sirobhushanam
School of Computing
Clemson University
Clemson, USA
ssirobh@g.clemson.edu

I. Abstract

Anomalous samples in data set cause hindrance in recognition and classification. For textual data, anomalous patterns causes unnecessary bias,mis-classification and higher reconstruction error. For our project, we wish to detect anomalous patterns in consumer review of product and service. We selected publicly available dataset of Yelp Reviews,deceptive opinion spam corpus version 1.4 and amazon product review.Both Yelp and Deceptive corpus contain restaurant reviews collected from different zip code region located in North America collected through web portals from Yelp, TripAdvisor and crowd-sourced with Amazon Mechanical Turk. Amazon review corpus contains product reviews from different categories. Reviews have been labeled as deceptive or truthful which we used in our fake review classifier model. By classifying fake or spammed reviews as anomalous, those could be filtered out for future recommendation of said product or service and mining user opinion from those reviews. Deep neural network model with multiple layers of convolution and Long Short Term Memory (LSTM) were trained to distinguish between fake and original reviews. Subsequently we performed sentiment classification of truthful and deceptive reviews and observed the impact of anomalous review on the classification accuracy.

II. Introduction

Online reviews have played very important role over the course of recent times as the booming industry of e-commerce, search engines and social media networks grows larger than ever. As every day people purchase products online, visit restaurants or services and browse social media, they read and post online reviews of their experience. Online reviews have such an influence on peoples every day decision making that business of fake or deceptive review writing have emerged. Companies or services pay users to produce deceptive reviews to inuence readers opinions. These practices severely compromise the reliability and value of products and services. A classifier that would detect anomalies such as deceptive reviews would provide user with the belief and reliability. We propose several classifier models for deceptive review detection that leverage machine learning and deep learning based algorithms. Our semi-supervised classifier catches fake reviews with nearly 90% prediction accuracy where the model we trained with fairly limited number of labeled deceptive reviews. We generalized models to extract features from various sources of online review and learn to recognize the patterns of deceptive reviews. We performed opinion mining on classified reviews to estimate positive or negative sentiment and analyzed if detection of deceptive reviews had impact on sentiment classification accuracy.

III. Related Work

Companies such as Yelp and Google have their own fake review detection algorithm deployed. Yelp hands out retribution to businesses caught with fake reviews like ranking penalty, prohibit advertising, ban and put under monitoring. Google apparently does not and their review filter is not automatic either. Many state of the art review detection approaches utilize reviewer centered features i.e average review length, time series analysis to monitor review dates, ratings deviation. Some exploits review centered features such as ngram features, semantic features, structure of sentences in the review Wang et al. [11]. Few approaches have used time series analysis to detect anomalies in the posting timeline of reviews. Dhingra et al. [14] uses unsupervised fuzzy clustering method to detect opinion spam detection which showed about 80% accuracy

IV. Proposed Approach

Our main focus is to detect and filter out fake,spam or deceptive review. Reviews usually contain product and user information, timestamp, ratings, and text. All those attributes of review can be considered as features and mined to determine the authenticity of reviews. Anomaly in timestamp of review would require time series analysis. For anomalies in review text itself, language properties would need to be modeled. Our primary choices for review is regional restaurant review data set from Yelp, hotel data from TripAdvisor etc. and crowd-sourced data from Amazon Mechanical Turk and product review data set from Amazon. Spammer or deceptive reviewers would give false positive opinions to some target service or product, in order to promote the entities and/or by giving false negative opinions to others in order to damage their reputations. It can also include advertising spam or bot generated reviews. Our aim is to identify and classify them with semi supervised deep learning based approach. We collected online review corpus from various platform such as Yelp, TripAdvisor, e-commerce giant Amazon. Partial data that we collected was labeled with deceptiveness factor, others we labeled the data based on structural, latent features in the review text. Our models performed well for all 3 source of labeled fake review data and generalized the feature extraction process to adapt to most online review corpus. We propose several neural network based models for fake review detection and they demonstrate quite a bit speedup of training time. Our support vector machine based classifier performed remarkably well as neural network models. We analyzed the effect of deceptive review detection on opinion mining and sentiment classification. Separating deceptive and true reviews enabled us to mine opinions on non-spam or non malicious reviews and increased the accuracy by significant margin.

V. Data Preprocessing

A. Dataset Description

- Yelp.com has made user and business information available to public as part of yelp open challenge [https://www.yelp.com/dataset]. Yelp has also released restaurant review data that can be used for outlier detection. Reviews were collected from various regions in several states of North America such as NY, NJ, VT, CT, and PA. Reviews include product and user information, timestamp, ratings, and a plaintext review. Yelp Chicago, NYC and Combined regions have 67,395, 359,052 and 608,598 reviews of hotels and restaurants, respectively. These have been made available here [http://odds.cs.stonybrook.edu/yelpchi-dataset/] and first used by Rayana and Akogulo [5] [6], Mukherjee et al. [7]. With help of Yelp anti fraud filter, datasets contain both recommended and filtered reviews considered to be genuine and falsely generated respectively. The review dataset labeled with Yes(True) or No(False) help us train a model that predicts if a given review is fake or not. Rayana et al. [5] released a compact, cleaned review text data and metadata information and allowed our team to use the dataset with proper citation [http://http://bit.ly/2PknY8i].
- Deceptive opinion spam corpus version 1.4 [8] contains 400 truthful positive reviews from TripAdvisor, 400 deceptive positive reviews from Mechanical Turk, 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp,400 deceptive negative reviews from Mechanical Turk. Each set contains 20 reviews for each of the 20 most popular Chicago hotels.
- Amazon review corpus contains products reviews with attributes such as review id, labels denoting truthful or deceptive, rating, verified purchased, product id, product title, review title and review text. Dataset was collected upon request from Dr. Bing Liu[]. Raw data contained 5838923 records which we sampled to get subset of 21000 samples and annotated the reviews with labels. This dataset has been used in [1], [4], [12], [13].

B. Data Transformation

1) Parsing: Review, business,user of Yelp datasetwas formatted as Json object file which was transformed to pandas dataframe for parsing the features. Data attributes include business id, review id, user id, funnyness,usefulness,coolness factor of the business,text body and rating value.Sample review records are shown in Fig. 1. We filtered out only restaurant reviews from all other businesses for outlier detection. We ignored closed restaurant reviews. When we visualized the mapping between the review length and the star ratings, we observed the distribution of text length is similar across all five ratings in Fig. 2. However, the number of text reviews seems to be skewed a lot higher towards the 5-star ratings. A box plot view of the text length for each star rating indicates text length may not be correlated with rating in Fig. 3.
Opinion spam corupus contained truthful and deceptive reviews with positive and negative polarity. Each review and corresponding label was stored as individual file. Each review had average 158 words.
Amazon review corpus was parsed into dataframe for cleaning duplicate reviews and ratings with NaN values as shown in Fig. 4. Grouping product for each member is illustrated in Fig. 5

2) Cleaning: For both Yelp and Opinion spam corpus, we selected textual review as input data and labels categorized as deceptive or true as output label for classification. Only English reviews were preserved by help of LangDetect library.Subsequently , review texts were cleaned of unknown characters, punctuation, whitespace characters. Common stopwords and sparse terms were also removed and converted to lowercase. Stemming of words to their word base and lemmatization were performed using available libraries from numpy and spaCy. Review that had missing labels were dropped. For Amazon review we dropped non english review, removed stopwords, punctuation,unknown characters, lemmatized and created bigrams to be used later in support vector machine model of deception detection. Word Punctuation tokenzier from Microsoft NLTK library helped us detect all known stopwords in English.

C. Data Augmentation

Yelp outlier detection dataset provided by Rayana et al. [5] [6] were imbalanced by more true reviews than deceptive reviews. So we augmented the dataset by duplicating fake reviews to balance with the true reviews. We also created a word level recurrent neural network model to generate word sequences based on learned parameters, given a corpus of deceptive review text only.

D. Vectorization

We used tokenizer module from Keras utilities to create feature vectors from text. Module transformed text data to numerical vector by assigning numerical rank in order of most to least common words.We add padding to make all the vectors of same length that is maximum sentence length of 500. Padding is necessary for use in convolution later. Tokenizer converted the text to integer indexes which could

	business_id	cool	date	funny	review_id	stars	text	useful
0	ujmEBvifdJM6h6RLv4wQIg	0	2013-05-07 04:34:36	1	Q1sbwvVQXV2734tPgoKj4Q	1.0	Total bill for this horrible service? Over $8G...	6
1	NZnhc2sEQy3RmzKTZnqtwQ	0	2017-01-14 21:30:33	0	GJXCdrto3ASJOqKeVWPi6Q	5.0	I *adore* Travis at the Hard Rock's new Kelly ...	0
2	WTqjgwHlXbSFevF32_DJVw	0	2016-11-09 20:09:03	0	2TzJjDVDEuAW6MR5Vuc1ug	5.0	I have to say that this office really has it t...	3
3	ikCg8xy5JIg_NGPx-MSIDA	0	2018-01-09 20:56:38	0	yi0R0Ugj_xUx_Nek0-_Qig	5.0	Went in for a lunch. Steak sandwich was delici...	0

Fig. 1. Sample Yelp Review records

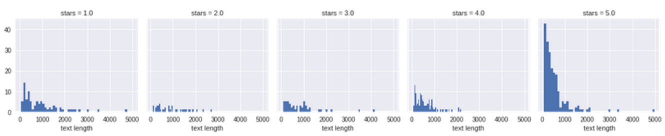

Fig. 2. Distribution of text length

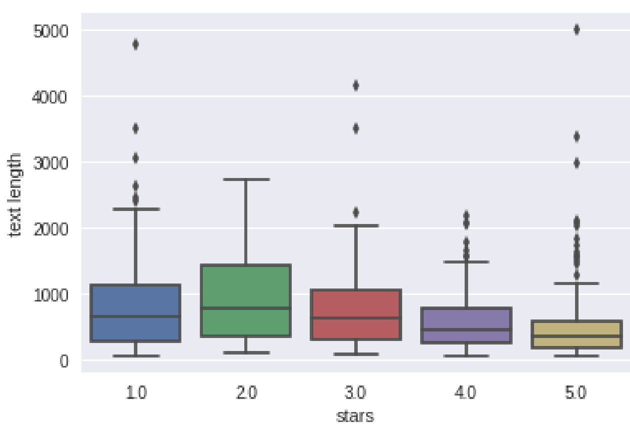

Fig. 3. Box plot of text length for each rating

be passed on to embedding layer. We found 147858 unique tokens.As result, we have formatted text samples and labels into tensors that can be fed into fake review detection neural network as input.

We chose Tf-Idf vectorization for feature extraction to be used for fake review detection support vector machine model. Feature vectors are sparse matrix where each word corresponds to its normalized frequency in review. Vectorizer used bigrams and minimum frequency of 3 words as threshold values.

VI. Fake Review Classification Model: Neural Network

A. Network layers

After data cleaning and tokenization, we built our model one layer at at time. We experimented several models with different configuration of hidden layers, while maintaining the same input and output layer for each dataset.

	member id	product id	date	number of helpful feedbacks	number of feedbacks	rating	title	body
0	A1004AX2J2HXGL	B0007RT9LC	May 30, 2005	3	4	5.0	The film speaks for itself	The only thing missing is a presentation of t...
1	A1004AX2J2HXGL	B00028HBKM	January 17, 2005	9	22	1.0	"Life beyond the thunderdome"	I cried when I saw The PASSION OF THE CHRIST ...
2	A1004AX2J2HXGL	B00062IVM6	January 17, 2005	4	7	5.0	I'm making my lunch!	WILD AT HEART on DVD. I've been waiting on t...
3	A1004AX2J2HXGL	B00064LJVE	January 13, 2005	5	15	1.0	Into the woods	M. is a hack, a second-place magician in a hi...
4	A1004AX2J2HXGL	B0002GMSC0	January 6, 2005	15	15	1.0	koo-koo-ca-choo	Where do people come up with this garbage? ...

Fig. 4. Amazon review dataset

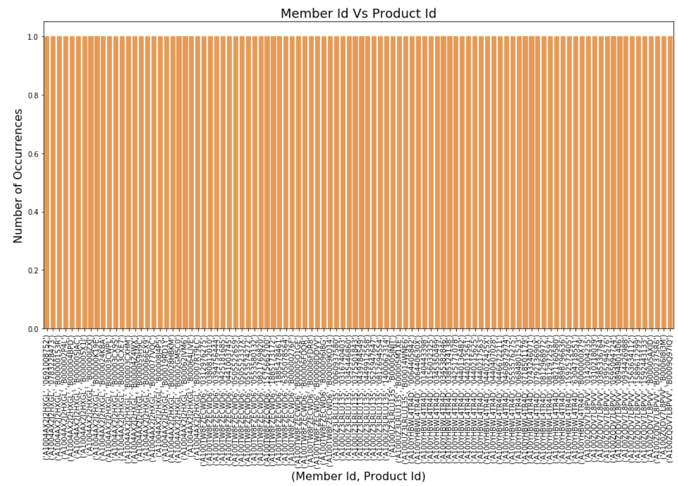

Fig. 5. Member Id vs. Product Id

Input Layer: Input layer takes input data in shape of (batch size, maximum sequence length). We chose maximum sequence length of 500 for the Yelp review dataset and length of 100 for deceptive review corpus.

Embedding Layer: Word Embedding constructs the vector representation of words where common and most important terms are placed together. Word embeddings are learned while training a neural network on the classification problem. Embedding layer stores a lookup table to map the words represented by numeric indexes to their correct dense vector representations. Size of embedded vector impacts the training time.Usually size is correlated with maximum length of input sequence. We used two different word embeddings for our project. One of them is GloVe- Global Vectors for Word Representation made available by Stanford NLP group [9] https://nlp.stanford.edu/projects/glove/. Global vector representations were trained by aggregating global word-word co-occurrence statistics from a corpus and projected into linear word-vector space. We loaded pre-trained Glove embeddings and experimented with both keeping embeddings fixed or non-trainable and trainable. Keeping embeddings fixed reduced training time whereas training the embeddings gave slightly better accuracy, for certain cases. Another pre-trained embeddings we used is Google's Word2Vec,which includes word vectors for a vocabulary of 3 million words and phrases that they trained on roughly 100 billion words from Google News. The vector length is 300 features. We applied gensim library to load pre-trained vectors. Compared to when we used one hot encoding from continuous bag of words(CBOW) vector model, using pre-trained embeddings made our classification task easier. In short, input of embedding layer should have vocabulary size, the size of the real-valued vector space as embedding dimension, and the maximum length of input documents. Output of embedding layers is 3D tensor of shape (samples, sequence_length = 500, embedding_dim = 100).

Hidden Layers: Our choice of hidden layers included sequential layers of 1 dimensional convolution layers, 1 dimensional max pooling, global max pooling layer, fully connected dense layers, dropout layers, unidirectional and bidirectional long-short-term-memory layers. We did several experiments on optimal sequence of layers, kernel size, activation function and found two combination of hidden layers that achieved highest accuracy,compared to simple Recurrent Neural network or single convolution layer. We denote them as hidden_config_one an hidden_config_two.

Hidden_config_one has one 1 dimensional convolution layer which takes 500 length sequences of 128-dimensional vectors as input. So the layer had 128 kernel filters, each of size 5 by 5 to extract features and uses Rectified Linear Unit as activation function. Next layers is max pooling for purpose of subsampling. For pooling size of 5, we look at 5 word vectors at a time and take their maximum to create one uni-variate feature vector. This pair of convolution and max pooling layer is repeated once more. Then it was followed by another 1D convolution layers to get higher dimensional features which produced output of shape (batch size,steps,vector size). Output was passed on to Global Max pooling layers which takes maximum over the time steps to produce output of shape (batch size, vector size). This layer was followed by fully connected dense layer to propagate outputs to another dense layer with sigmoid activation for classification.
By using mutiplle layers of convolution with multiple kernel filter size and concatenating their outputs, we are detecting patterns in text with sliding window size of 2,3,5 adjacent words. Convolution layers are useful for detecting words regardless of their position in the text.

Hidden_config_two is based on hierarchical attention based network proposed by Yang et al. [10]. This network exploits hierarchy of sentence,words,characters in a text and applies two different attention mechanism at word level and sentence level. By focusing attention to words that contribute more to meaning of sentence, word attention vector is constructed. Same way, sentences that clue in the meaning of review text are paid more attention to create context vector. The input for this network is reshaped to become a 3D tensor of shape (batch size, sentences count , word count of each sentence).As such, we need to encode individual sentences before we encode the full text. Sentence encoder takes input of shape (batch size, maximum word count of each sentence) and passes it through embedding and Bidirectional LSTM layer. Bi-directional LSTM to read words in a sentence as a sequence and apply attention for their position in the sentence. Document encoder takes tensor of shape (maximum sentence count in document, maximum word count of each sentence). Keras has a really useful layer called Time Distributed layer is which allows the model to run a copy of the sentence encoder to every sentence in the document. Output of sentence encoder is fed through a bi-directional LSTM with fully connected connected on top to connect to the output softmax or sigmoid layer.

Output Layer: For our models, we alternatively chose binary labels and categorical labels for classification. Binary labels were 0 for deceptive and 1 for original/true in categorical label classes, respectively. We initially chose Softmax function for classification, which is essentially a multi-class logistic regression. It returns the probabilities of each class and the target class will have the high probability. Since our task was binary classification, we replaced the softmax function with sigmoid function and fitted the model keeping other parameters fixed.

Dropout and batch normalization: Adding a dropout layer after embedding or before softmax layer is considered a regularization technique, that prevents model from overfitting and increases the validation accuracy. Dropping a few neurons from the layer with a certain rate makes the model generalize better against validation set. We observed nearly same training and validation accuracy for Yelp dataset after we added a dropout layer with 20% rate , followed by embedding layer.

Loss function and optimizer: We selected both binary cross-entropy and categorical cross-entropy for loss function. We experimented with "focal loss" function typically reserved for multi-class classification task on imbalanced dataset. For imbalanced dataset, training is inefficient and models can be over-fitted. Focal loss functions decrease the weights for inliers or easy examples and is best for sparse set of training data. It requires a parameter, gamma, to decide the rate of adjustment to down weight the samples. However, it didn't improve the validation accuracy much. We alternatively chose ADAM, Root Mean Square and Gradient Descent optimizer to train the data. Root mean square and gradient descent optimizer performed best for deceptive opinion corpus and amazon review corpus. Smaller learning rate as 0.0008 caused slower convergence but increased validation accuracy.

B. *Train and validation set preparation*

The set of reviews as sequence of sentences was prepared as the input features and corresponding label set denoting deceptive/truthfulness of review as target class. Reviews being of uneven length, they were padded to maximum length of 500. Typically this parameters, maximum length of input sequence would be chosen as the vector size for the embedding layer. We experimented with different values of validation split ratio of input dataset, namely 0.2,0.6 and 0.8. We split the data into train and validation set by random selection for each batch. While training models, we feed the input features and labels by generating batches because it prevented resource allocation and out-of-memory issues. We simultaneously train the model on training set and cross validate it on validation set.

C. Prediction on test set

We created a test subset separate from the training and validation set, from the original Yelp open dataset. Models were used to predict the deceptiveness or truthfulness of unlabeled reviews. We calculated accuracy of classification based on ratio of predicted versus true labels.

D. Batch generation

Training the model by generating batches of review and corresponding label improved the training efficiency for larger number of epochs and solved the memory shortage issue we encountered. We followed keras tutorial to generate data items on the fly given shuffle set of labels and review texts.

VII. Fake Review Classification Model: Support Vector Machine

A. Yelp Dataset

We wanted to compare our neural network model with a support vector machine widely recognized for classification purpose.We used the feature extraction module called TfidfVectorizer which is used to transform each review to large sparse matrix each cell representing a word and the frequency at which it appeared and then perform normalization. Feature vectors obtained from the previously described step were passed on to linear support vector based classification which works well for sparse input matrix. We considered bigrams only and have removed all the words that appear less than 3 times to avoid noise.

B. Amazon Dataset

Amazon datasest showed marginally poor validation accuracy for model_1 and model_2 , less than 80%. Support vector machine performed a lot better when we incorporated features like verified purchase of product, duplicate reviews from same user with the original deceptive label and review text. Validation accuracy was increased to 81% by including those features.

VIII. Opinion Mining

After we successfully classified text reviews as true or deceptive, we analyzed the impact of detection on mining opinion and sentiment from classified reviews. We expected that estimation of positive or negative opinion would be much easier to predict if the deceptive reviews are removed. To mine polarity of opinion from the reviews and classify as positive or negatively biased, we implemented several classification models. We selected several subsets from original review. We used Count Vectorization to convert the text collection into a matrix of token counts.The resulting matrix would be 2-D matrix where each row is unique word and each column is a review. We chose one-hot-encoding to encode ratings of review to numeric values. We considered only ratings with review 1 and 5.

TABLE I
Model1 classification report(Yelp)

	Precision	Recall	F1-score	Support
0	0.75	0.94	0.83	80510
1	0.94	0.79	0.85	105550
micro avg	0.83	0.85	0.84	186060
macro avg	0.84	0.86	0.84	186060
weighted avg	0.85	0.85	0.84	186060
samples avg	0.84	0.85	0.84	186060

A. Logistic Regression

Then feature vectors were trained with Logistic regression model and predicted on the test data.

B. Multinomial Naive Bayes

We also trained on a MultiNomial Naive Bayes model. For both model, parameters values were selected optimally through exhaustive search over all possible values. We have tuned the parameters of the model such as penalty to L2 and predicted new review with accuracy of 98%.

IX. Experimental Results

A. Fake review detection

Our balanced fake review dataset from Yelp had 486767 samples for training and 121691 samples for validation. By using Glove pre-trained word embedding vectors, we found 204757 unique tokens and 400000 word vectors. Size of input tensor for review text was 608458 by 500 having maximum sentence length 500 and label tensor of shape 608458 by 2 for 2 classes. We named the neural network based fake review detection model with hidden_config_one as model_1, hidden_config_two as model_2 as shown in fig.15 and fig.8 accordingly. We trained all of our models for training epoch of 10 and batch size of 128. Size of trainable parameters significantly reduced when we used pre-trained word embeddings. Adding a dropout layer after embedding layer in model_1 changed the validation accuracy to 84%. TableI and TableII shows the classification report from model_1 and model_2 of fake review detection on Yelp dataset. Loss curves and accuracy curves for model_1 and model_2are plotted in fig.6,fig.7,fig.10,fig.11 respectively. With SVM, we have obtained accuracy about 91.5% for Yelp classification dataset as shown in Fig.9. Classifcation report was shown in III. Comparison of performances for 3 different models of fake review detection in metrics of training accuracy, validation accuracy and accuracy_score is shown in tableIV.

B. Opinion mining

Subset of data we used for opinion mining on Yelp has 2118439 samples of training and 907903 samples for validation. Parameter for score used in exhaustive grid search was around 0.93. We used 10 fold cross validation for both the naive bayes and logistic regression model. ROC(Receiver Operating Characteristics) curves for amazon dataset on naive bayes and logistic regression are shown in fig.13 and fig.14. ROC curve for Yelp dataset on Naive Bayes is shown in fig.12.

TABLE II
MODEL2 CLASSIFICATION REPORT(YELP)

	Precision	Recall	F1-score	Support
0	0.73	0.84	0.78	80517
1	0.86	0.76	0.81	105543
micro avg	0.80	0.80	0.80	186060
macro avg	0.8	0.92	0.797	186060
weighted avg	0.8	0.92	0.797	186060
weighted avg	0.8	0.92	0.797	186060

TABLE III
SVM CLASSIFICATION REPORT(YELP)

	Precision	Recall	F1-score	Support
0	0.86	0.97	0.91	121020
1	0.97	0.88	0.92	158071
avg/total	0.92	0.92	0.92	279091

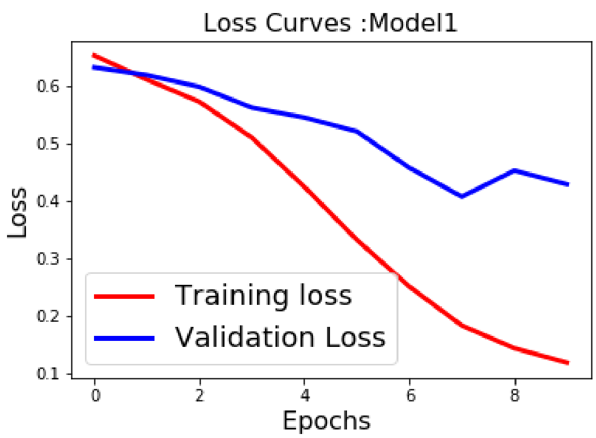

Fig. 6. Loss Curves : model_1

X. CONCLUSION

In this paper, we mined several publicly available online review corpus and analyzed several features of the text review. We performed two data mining approaches,one of them is anomaly and outlier detection from textual review and other is opinion mining from text reviews.We employed semi supervised approach with relatively small amount of deceptive review dataset compared to original review data. Our approach classifies reviews as true or deceptive with several neural network model and achieves validation accuracy in range of

TABLE IV
PERFORMANCE COMPARISON: FAKE REVIEW DETECTION

Dataset	Model	Acc(Train)	Acc(Val)	Acc_Score
Yelp	model_1	0.8	0.92	0.797
Yelp	model_2	0.8	0.92	0.797
Yelp	svm	0.8	0.92	0.797
Decision corpus	model_1	0.93	0.803	0.7
Decision corpus	model_2	0.98	0.75	0.75
Amazon	model_1	0.99	0.62	0.62
Amazon	svm	0.82	0.81	0.81

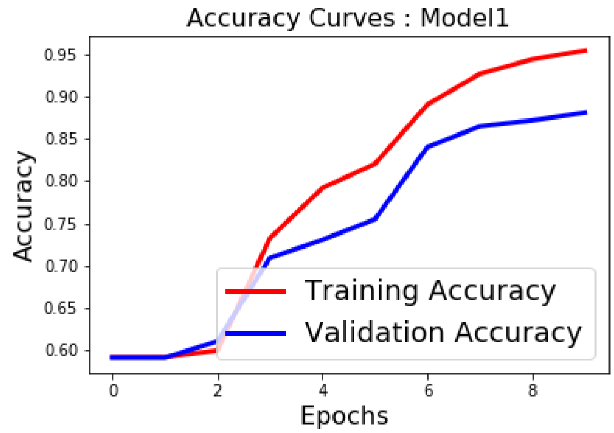

Fig. 7. Accuracy Curves : model_1

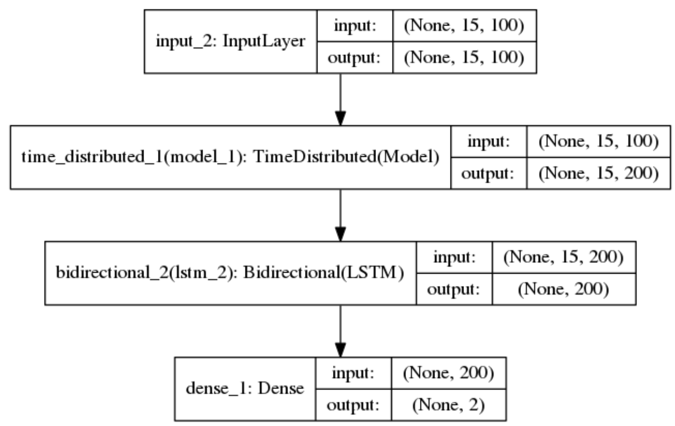

Fig. 8. Fake review detection model_2

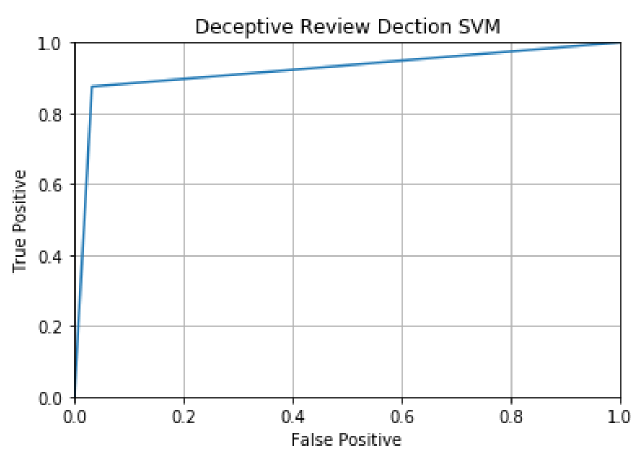

Fig. 9. ROC curve: Mining opinion on Yelp review

80-90%. Through detecting truthfulness of review, we were able to execute opinion mining on reviews more efficiently and it achieved better positive or negative classification accuracy. Our approach was able to generalize classification for multiple online review corpus such as Yelp, deceptive opinion corpus

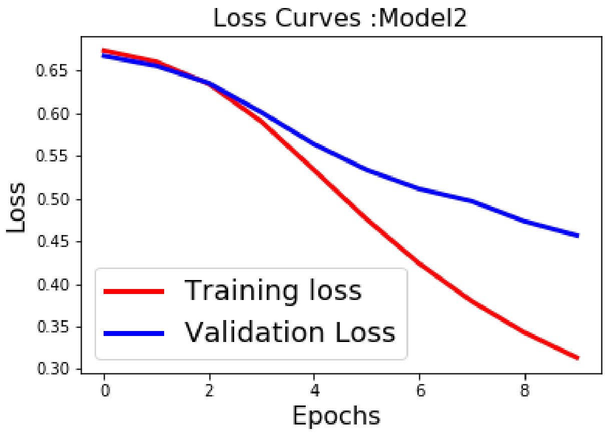

Fig. 10. Loss Curves : model_2

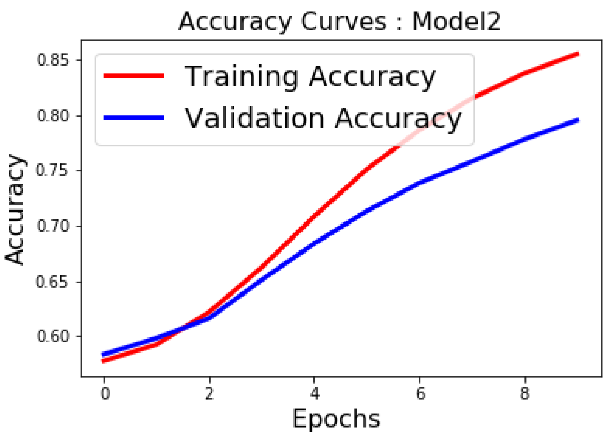

Fig. 11. Accuracy Curves : model_2

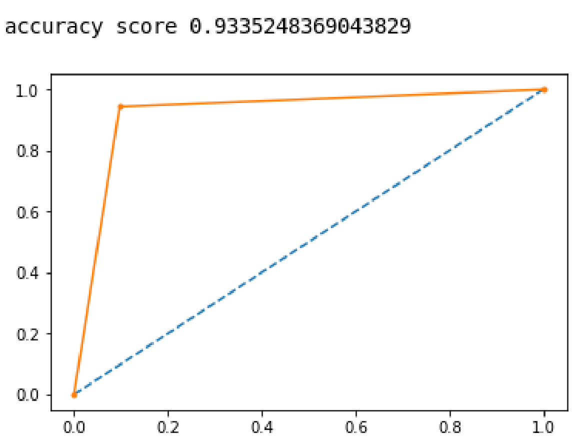

Fig. 12. ROC curve:Yelp opinion mining(Naive Bayes)

and Amazon product review.

REFERENCES

[1] Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee and Jidong Shao. Bimodal Distribution and Co-Bursting in Review Spam Detection. In proceedings of the ACM International World Wide Web Conference (WWW'17). April 3-7, 2017, Perth, Australia
[2] Santosh Kc and Arjun Mukherjee. On the Temporal Dynamics of Opinion Spamming: Case Studies on Yelp. In proceedings of the ACM International World Wide Web Conference (WWW'16). April 11-15, 2016, Montreal, Canada.
[3] Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Jidong Shao. Analyzing and Detecting Opinion Spam on a Large-scale Dataset via Temporal and Spatial Patterns. In proceedings of the 9th International AAAI Conference on Web and Social Media (AAAI-ICWSM'15). May 26-29, 2015 Oxford, UK.
[4] Zhiyuan Chen, Nianzu Ma and Bing Liu. Lifelong Learning for Sentiment Classification. Proceedings of the 53st Annual Meeting of the Association for Computational Linguistics (ACL-2015, short paper), 26-31, July 2015, Beijing, China.
[5] Shebuti Rayana, Leman Akoglu.Collective Opinion Spam Detection: Bridging Review Networks and Metadata. , ACM SIGKDD,August 10-13, 2015 , Sydney, Australia.
[6] Shebuti Rayana, Leman Akoglu, Collective Opinion Spam Detection using Active Inference. , SIAM SDM, May 5-7, 2016, Miami, Florida, USA.
[7] A. Mukherjee, V. Venkataraman, B. Liu, and N. S. Glance,What Yelp fake review filter might be doing? ICWSM, 2013.
[8] M. Ott, Y. Choi, C. Cardie, and J.T. Hancock.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,2011.
[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation, 2014.

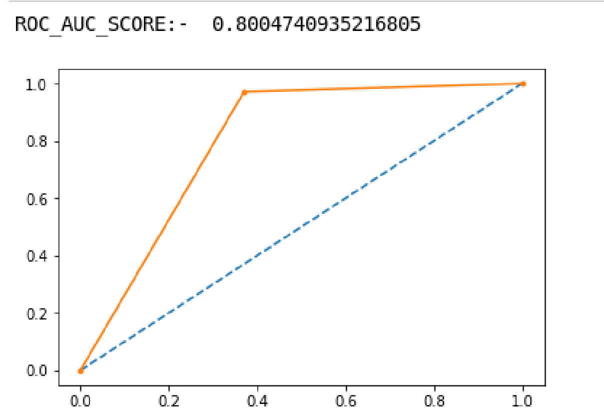

Fig. 13. ROC curve : Amazon opinion mining(Naive Bayes)

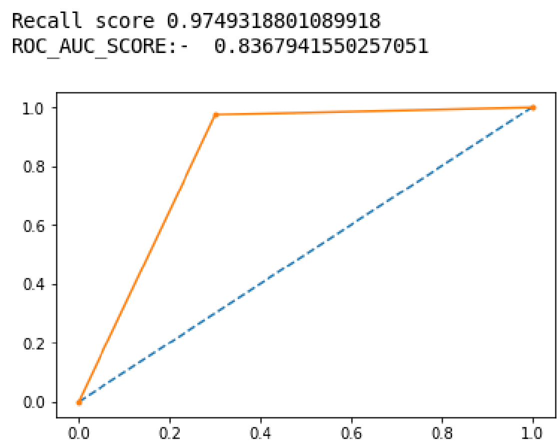

Fig. 14. ROC curve : Amazon opinion mining(Logistic Regression)

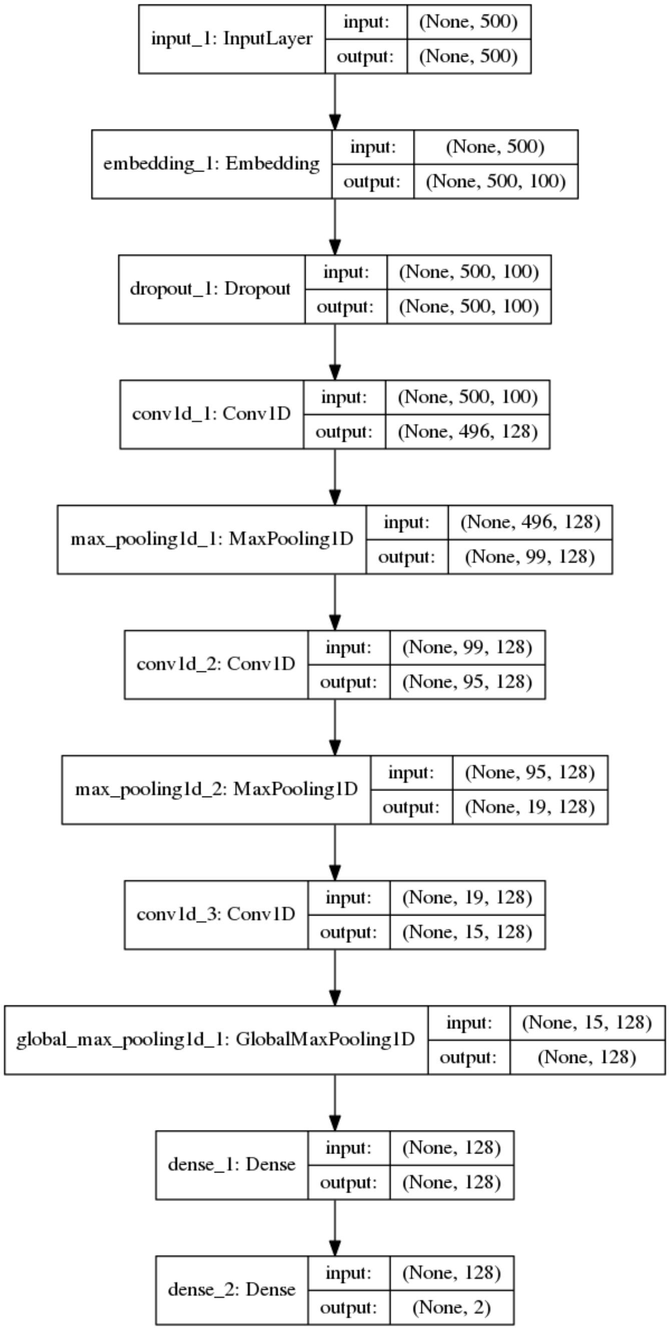

Fig. 15. Fake review detection model_1

[10] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,2016.

[11] Wang, Zehui, Yuzhu Zhang and Tianpei Qian. Fake Review Detection on Yelp",2017.

[12] Nitin Jindal, Bing Liu, Opinion spam and analysis, WSDM, 2018.

[13] Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei and Jidong Shao, Spotting Fake Reviews via Collective Positive-Unlabeled Learning, ICDM, 2014.

[14] Komal Dhingra,Sumit Kr Yadav, Spam analysis of big reviews dataset using Fuzzy Ranking Evaluation Algorithm and Hadoop, International Journal of Machine Learning and Cybernetics, 2017.