Group 6: Can fake news detection models also perform fake review detection?

Abhishek Saini Ankita Dey Debanjan Chatterjee
Gautam Chauhan Shilpa Chatterjee
20111401, 20111013, 20111016, 20111020, 20111057
{abhik20, ankitadey20, debanjan20, gautamc20, shilpa20}@iitk.ac.in
Indian Institute of Technology Kanpur (IIT Kanpur)

Abstract

In this Project, we aim to study how well state-of-the art Fake News Detection systems work on a similar application, Fake Review Detection. We mainly focus on systems that uses Stance Detection as a major component to differentiate real and fake news. We use an Yelp Review Dataset for our project and pass the preprocessed data through 7 different models in various areas: CNN with LSTM, bi-LSTM (with and without network features), Siamese model (with and without network features), Multi-layer perceptron and LSTM(with one hot encoding). Models using Passive-Aggressive classifier, XGBoost, multinomial Naive Bayes classifiers has also been explored. We then evaluate each of the model's performance using macro F1 score and accuracy and analyse the outcomes.

1 Introduction

In recent times, online reviews have considerable control over user's buying decisions, but fake news is becoming a plague. Wikipedia defines fake news as 'Fake news is false or misleading information presented as news. It often has the aim of damaging the reputation of a person or entity, or making money through advertising revenue.' Fake Reviews similarly, are reviews by people who didn't use the product or service they are reviewing. It can be written by individuals who are paid by competing companies to write fraudulent reviews or even generated by bots, or can be written by individuals or employees of a company for personal vendetta. Fake Reviews can mislead consumers and deteriorate standing of deserving business in spite of no fault of their own.

While the problem of fake news is a well established problem with many papers, all with their unique approaches to solve the problem, there hasn't been much work on fake review detection specially using Stance Prediction, which is a key method used for Fake News detection. In this project, we attempt to implement some state-of-the-art fake news detection approaches that emphasises on Stance detection, on Yelp Review dataset and evaluate how well fake news detection models work on Fake Review dataset.

In section 2, we introduce some related work on this area. Next in section 3, we explore ideas of using Stance Prediction for fake news detection which has been our primary motivation behind taking up this research area. In section 4 we introduce the dataset used in our experiments, followed by the overview of experiments we conducted in section 5. Towards the end, we talk about the results obtained and error analysis in section 6 and 7. We wrap up the report with individual contribution and conclusion in sections 8 and 9 respectively. All of our models are available at `https://github.com/gautamchauhan04/fake-review-detection-models`.

2 Related Work

Although Fake News Detection is a major application of Stance Prediction, we couldn't find any paper that made use of Stance Prediction to detect Fake reviews. In fact there has also been a competition

on Fake News detection[1] in 2017 and more in the following years. In fake news detection, the task is to detect the Stance of the body of news article with respect to the heading. There are tons of papers on this dataset alone. After reading numerous papers on Fake News Detection especially using Stance Prediction, we decided to take up this area for our project but centered on Fake Reviews Detection instead of Fake News.

3 Proposed Idea

We have tried approaches which have performed appreciably for Fake News Detection with the help of Stance Prediction and evaluated how well those approaches work on Fake Review Detection. In general, if the stance of the body is related to the heading, that is, the body either agrees, disagrees or is neutral towards the body but isn't unrelated, it is considered to be not fake or real; similarly if the stance of the body is unrelated to the heading, the review is considered to be fake.

In [1], an end-to-end Stance Detection system that claimed third position in the Fake News Challenge, FNC-1 competition is presented. In the model, lexical (Term frequency of heading and body) and similarity features (cosine similarity between the l_2-normalised term frequency-inverse document frequency of the headline and body) are passed to a Multi-layer perceptron or MLP with just one hidden layer. Cross entropy between the system's softmax probabilities and the true labels is minimized while model is trained and Adam optimizer has been used.

[2] used embedding layers, which fed embeddings into a bi-LSTM model, and its hidden state is feed to a multi-layer perceptron (MLP) followed by a log-SoftMax to predict as fake or not fake [3] used a siamese network, which is used for checking similarity between two images. In our case of fake review detection, we used it to detect stance similarity between head and body of the review, to classify it as fake or real. It uses bi-LSTM with attention instead of CNN, which have been used for images.

In [4], the authors proposed a deep ensemble technique to tackle fake news classification. For the task, the authors used a CNN (Convolutional Neural Network) for extracting features from the claims and supporting document given in the task and passed these features to a LSTM (Long Short Term Memory) whose output was further passed on a MultiLayer Perceptron (MLP) for the classification task.

Models like passive aggressive, XGBoost classifiers and Multinomial Naive Bayes have also risen to popularity in fake news detection [5, 6, 7]. Hence, models using passive-aggressive, extreme gradient boosting classifiers and Naive Bayes to perform fake review detection has also been proposed.

4 Dataset/Corpus

Dataset used is a labeled Yelp review dataset, available at Kaggle[2]. The original dataset includes User_id, Product_id, Rating, Date, Review, Label columns as shown in figure 1.a. The dataset has two labels, 1 signifying real or genuine reviews and -1(changed to 0 during dataset modification) for fake reviews. For the task, the dataset was modified by splitting the review into head and column as shown in figure 1.b. At first, a review was broken into sentences using the NLTK sentence tokenizer. Each sentence was then word tokenized using NLTK, then stopwords and symbols were removed from these tokens. To split a tokenized review into head and body, minimum number sentences with number of words more than three was made head and rest as body maintaining five minimum number of words. Review which was not long enough to split in this way were removed. Number of words comparison of head and body can be seen in figure 2. The advantage of splitting by sentences maintains the context of sentences. Some models also used network features. This includes normalized mean and median of rating of a product, mean and median of the rating given a user, rating given by the user for the product, and the number of words in head and body. The modified dataset decreased the number of samples, but the dataset was still skew so +1 label samples were undersampled.

[1] `http://www.fakenewschallenge.org/`

[2] `https://www.kaggle.com/abidmeeraj/yelp-labelled-dataset`

	User_id	Product_id	Rating	Date	Review	Label
0	923	0	3	12/8/2014	The food at snack is a selection of popular Gr...	-1
1	924	0	3	5/16/2013	This little place in Soho is wonderful. I had ...	-1
14	937	0	4	10/15/2014	Tiny little place, but very good food. Pastits...	1
15	938	0	5	10/10/2014	I had the braised lamb sandwich and was one of...	1

(a)Orignal Labeled Yelp! dataset

	Head	Body	Label	Features
0	food snack selection popular greek dishes	underwhelmed main courses tables sometimes har...	0	5.0 4.435897435897436 39 4.0 4.009523809523809...
1	little place soho wonderful	price shocked small serving soho staff little ...	0	3.0 3.0 1 4.0 4.0095238095238095 210 4 3
10	spot close job decided check lunch	ordered think may special roasted chicken vege...	1	4.0 4.0 2 4.0 4.0095238095238095 210 10 3
11	lunch friend lamb sandwich juicy flavorful	good thing friend shared otherwise think would...	1	4.0 4.0 11 4.0 4.0095238095238095 210 8 4

(b) Modified dataset

Figure 1: Dataset

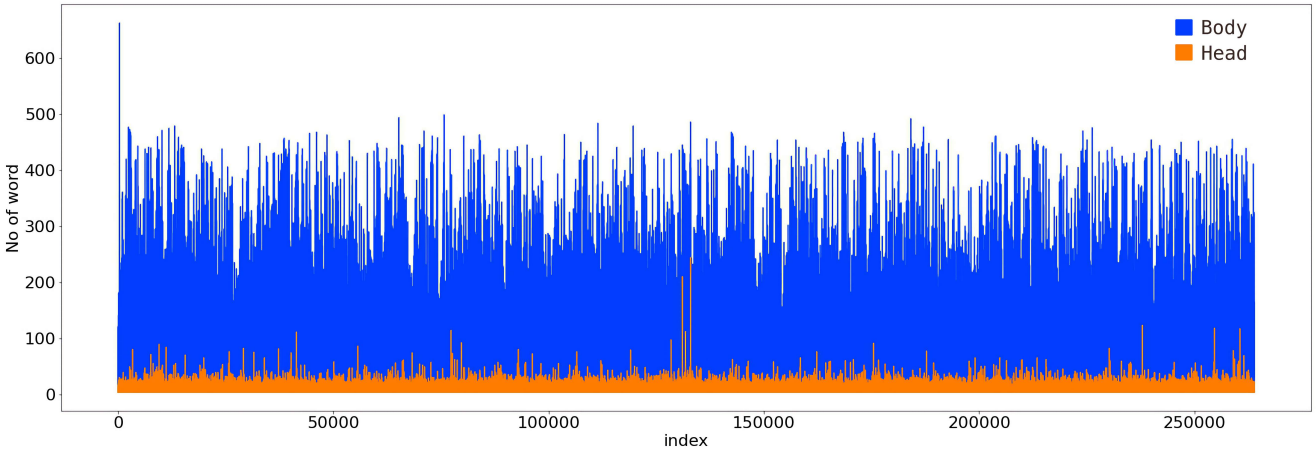
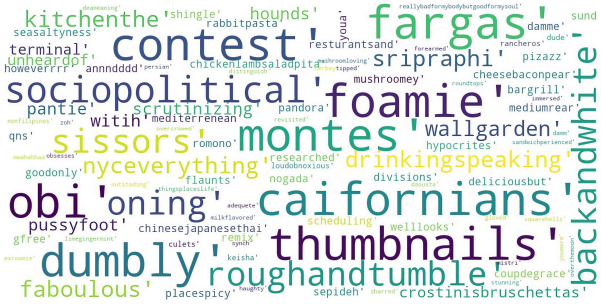
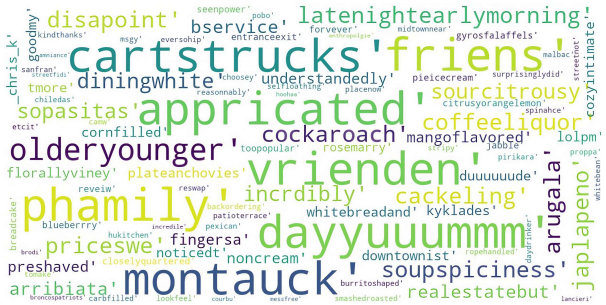


Figure 2: Comparison of number of words in head and body



(a) Corpus



(b) tokens(words) not in GloVe

Figure 3: Word cloud

5 Experiments

5.1 Ensemble of CNN and LSTM Model

Recognising good features manually to separate true from fake even for binary classification, is a difficult task for human beings. So, often people take help from deep neural models. We know that Convolutional Neural Network (CNN) is known to capture the hidden features efficiently. So, for this model, we use CNN to capture features from the head and body of the product reviews and these feature representations are then passed on to the Long Short Term Memory (LSTM) and the output of the LSTM is then passed on through a MultiLayer Perceptron to give us the output of whether the review is real or fake.The model was trained using binary-crossentropy loss and Adam optimizer with a initial learning rate of 0.001. Also, one must note that our dataset is highly skewed (with fake to real review ratio being 1:10 approx),so we have undersampled the majority class (i.e. the real reviews) to make the dataset balanced.

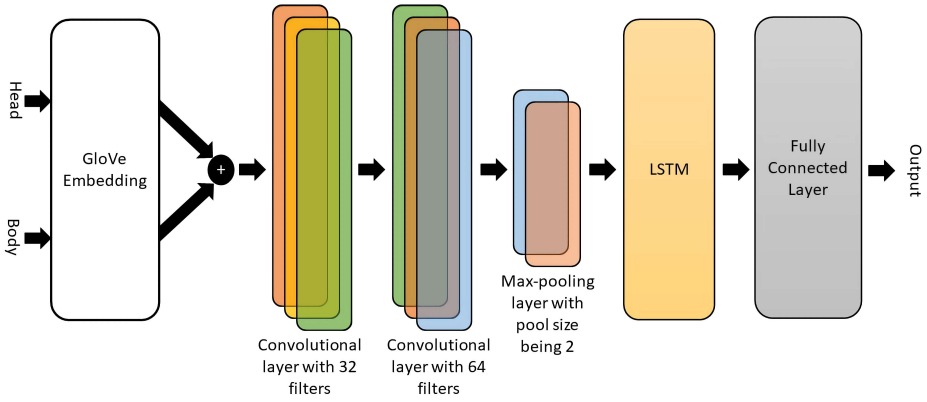


Figure 4: Model Architecture(CNN+LSTM)

5.2 Bi-LSTM

Figure 5 shows all the different components used for bi-LSTM and Siamese models. Bi-LSTM model uses an embedding layer, then a two-layer bi-LSTM model (figure 6.a), follower by a multi-layer perceptron (MLP) followed by a log-SoftMax (figure 6.b). For both head and body, the same embedding layer with an embedding size of 300 and the same two-layer bi-LSTM with the hidden size of 300. The hidden state of both head and body generated by bi-LSTM are concatenated with Manhattan distance and product of both, which are fed to MLP, as shown in figure 6.a without network features. In the bi-LSTM model with network features, network features were fed to a linear layer then given as input to the second layer of comparison MLP and the previous layer's output. Both bi-LSTM with and without network features were trained using Adam optimizer with a learning rate of 0.001, Cross-Entropy loss and batch size of 128.

5.3 Siamese model

Siamese model uses pre-trained glove embedding with two-layer bi-LSTM (figure 5.b) with attention (figure 5.c) then comparison MLP (figure 5.a). The same glove embedding of size 100, same two-layer bi-LSTM with a hidden size of 100, and the same attention is used for both head and body. Then, the output of both head and body from attention are concatenated with Manhattan distance and product of both, which is fed into a comparison MLP as shown in figure 6.b without network features. In the siamese model with network features, network features were fed to a linear layer then given as input to the second layer of comparison MLP and the previous layer's output. Both siamese models, with and without network features, were trained using Adam optimizer with a learning rate of 0.001, Cross-Entropy loss and batch size of 128.

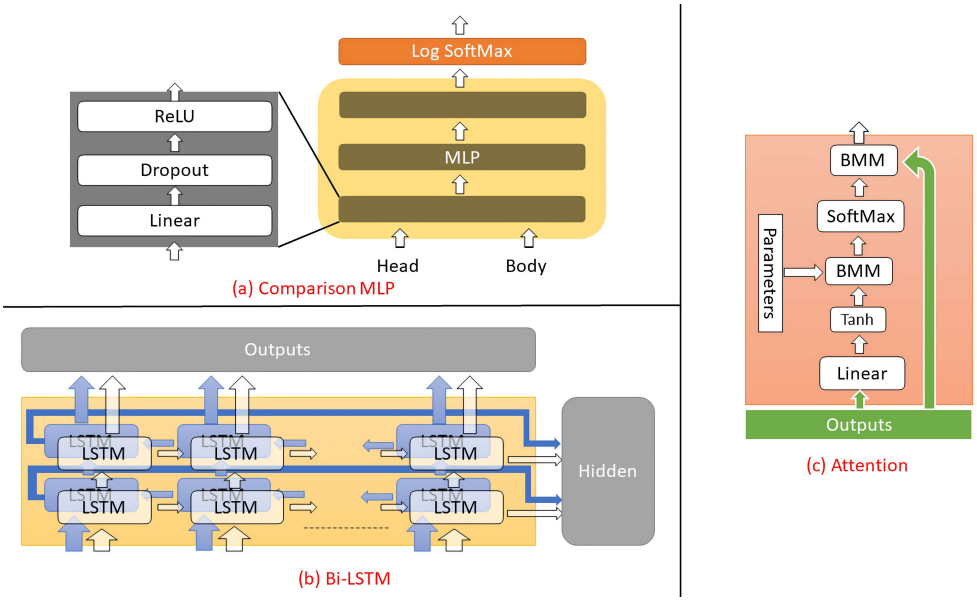


Figure 5: Model Components

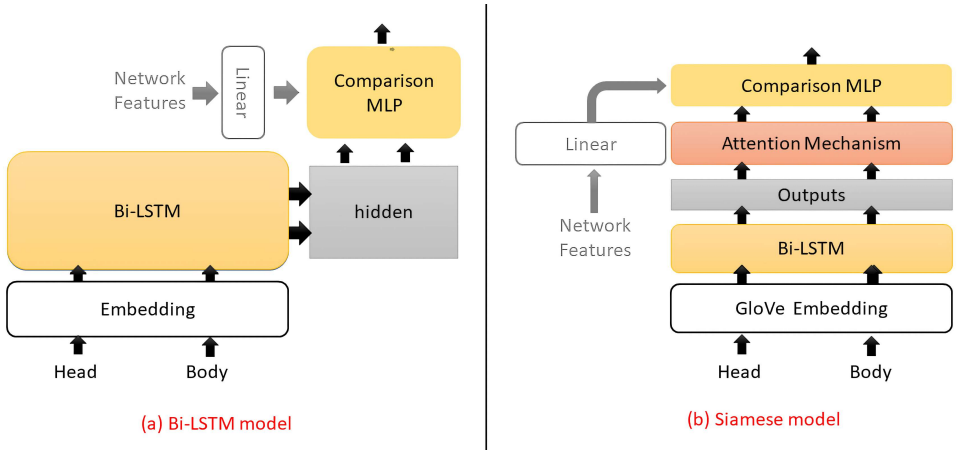


Figure 6: bi-LSTM and Siamese model

5.4 Multi-layer Perceptron

Since the original data is extremely skewed, the data is undersampled to get both the labels in ratio close to 1:1 and train-test split is done in ratio 80:20. With around 40,000 training data, 2 simple Bag of Words representation of text as said in Proposal idea is used. While [1] used 5000 most frequent words to create the Term Frequncy or TF vectors and term frequency-inverse document frequency or TF-IDF vector, resource contraint has forced us to use 3000 most frquent words to create bag-of-words (BOW) representations for the text inputs. These vectors are concatenated to get the feature vector which is passed to the MLP classifier. The classifier uses ReLU for non linearity in the hidden layers of 100 units. Softmax is applied to the output of final linear layer to obtain vector containing probability of each label. Finally the label with higher score is predicted. The model is trained in batches of size 500. The entire classifier and the additional functionalities are implemented using Tensorflow.

5.5 LSTM(With One Hot Encoding)

In this architecture, We used one hot vector encoding. First, we get Embedding of Headline and Body, and then we concatenate these to give input to LSTM, and then Sigmoid is applied to the output of

this for the Classification. The model uses a hidden layer of size 50, Adam optimizer, and categorical cross-entropy loss. Also, since the dataset is highly skewed, the data is undersampled to get labels in the same proportion, and the train test split is done in the ratio of 80:20 [8].

5.6 Models using Extreme Gradient Boosting and Passive Aggressive Classifiers

The text in the dataset corpus has been pre-processed using the above discussed techniques, such as removing stop words, normalization and lemmatization. The dataset is also highly skewed, hence under-sampling has been performed to achieve a class distribution of 1:1. A graphical illustration of the class distribution in the original and under-sampled dataset has been shown in Figure 7. For feature representation *term frequency-inverse document frequency* or TF-IDF has been used, which is a measure that evaluates how important a term is to a document in a collection of documents.

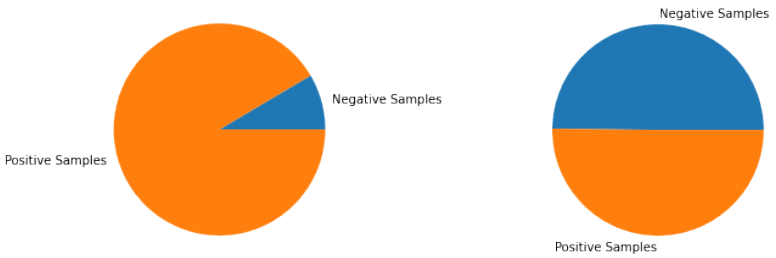


Figure 7: Plot showing the sample distribution in: (a) original dataset (b)undersampled dataset

Extreme gradient boosting (XGBoost): It is an ensemble learning technique based on gradient boosted decision trees, where errors made by the older models are corrected by adding new models. In gradient boosting the models are generated to predict the errors of the previous models, and they are added sequentially to make the final prediction. This process of adding models continues until no further improvements are possible [9].

Passive Aggressive Classifier: It is an online learning algorithm [10] that is used for large-scale learning. In, passive-aggressive classifier, the input data comes in sequential order and the model is updated step-by-step, as opposed to batch learning, where the entire train set is used at once. This can tackle a huge amount of data where it is infeasible to train on the entire dataset because of the large volume of data. Therefore the passive-aggressive classifier will use a training sample, update the model, and then discard that sample. It derives its name from the fact that the model is passive when the prediction is correct, that is, no change is made in the model. It is aggressive when the prediction made is incorrect, and thus the model is updated.

5.7 Model using Multinomial Naive Bayes

Multinomial Naive Bayes is a generative process that assumes; $P(x_1|y=k), P(x_2|y=k) \dots P(x_n|y=k)$ are independent given a class $P(y=k)$. Therefore, The joint probability of all the features conditioned on y=k is the product of each feature conditioned on $y=k$ [7].

$$P(x_1, x_2 \dots x_n|y=k) = \prod_{n=1}^{N} P(x_n|y=k)$$

Then, using Bayes Theorem, we can calculate the likelihood and prediction of new data. Also, Here order of words doesn't matter. To implement it, we used Positive and Negative examples in the same proportion to avoid the overfitting problem.

6 Results

The results obtained by different models on Fake Review detection task are as follows:

Model	macro F1 score	Accuracy
CNN with LSTM	**0.71**	**0.71**
bi-LSTM(without network features)	0.56	0.56
Siamese model(without network features)	0.59	0.60
bi-LSTM(with network features)	0.58	0.59
Siamese model(with network features)	0.61	0.61
Multi-layer perceptron	0.70	0.70
LSTM(with one hot encoding)	0.52	0.53
Multinomial Naive Bayes	0.70	0.70
Extreme Gradient Boosting	0.70	0.70
Passive Aggressive Classifier	**0.71**	**0.71**

Table 1: Macro F1 score and Accuracy of different models

From the results it is evident that models like Passive Aggressive Classifier, Extreme Gradient Boosting, Multinomial Naive Bayes, Multi-layer perceptron and Ensemble model (CNN with LSTM) provide us with the best accuracy and macro F1 score.

7 Error Analysis

In the CNN and LSTM model, it was observed that whenever the head or body of the review was too long, and had expressed strong emotions (like distress, joy,excitement), the model failed to remember the words that originally spoke about the product and gave more weight-age in remembering the emotions and ended up classifying the reviews wrongly.

In both the bi-LSTM and siamese model, the model with network features performed better than without network features. Network features can provide more context about the review in context with all other reviews giving by the user and all reviews given for a product.

In the paper [1], they have claimed an accuracy of 96.55%. Of course, accuracy alone can not be considered as a reliable metric specially in a skewed dataset (the FNC-1 dataset was highly skewed which has also been mentioned in several papers such as [11]). In the original skewed dataset of Yelp reviews that we used, this model was giving an accuracy of over 90% which shows that the model favours majority label in skewed dataset. However the F1 score was coming much lower to 50%. Using a balanced dataset, in which the skewed data is under-sampled to get both the binary labels close to 1:1 ratio, has reduced the accuracy but the F1 score, which is a much better metric to evaluate the model, has increased to 70.77%.

Multi Multinomial Naive Bayes gives 0.90 accuracy and 0.53 f1- macro score on original skewed data. However, after training with undersampled data, these values changes to 0.7 and 0.7, respectively.

Models using passive-aggressive and XGBoost classifiers also display a similar trend. When the model was trained on the original skewed dataset, the accuracy and macro F1 score for passive-aggressive classifier was 0.89 and 0.52 respectively, while the same for XGBoost was 0.48 and 0.92. However, after training the same models with the under-sampled dataset, both the accuracy and macro F1 score for the passive-aggressive classifier changes to 0.71, and for XGBoost it changes to 0.7. Given that the task at hand is to detect fake reviews, the macro F1 score is a more significant metric to judge the performance of the models, hence training with the balanced dataset produces better fake review detecting models.

8 Individual Contribution

Member Name	Contribution
Abhishek Saini	Pre-processing Dataset,LSTM(with one hot Encoding),Model Using Multinomial Naive Bayes(Using TfidfVectorizer)
Ankita Dey	Pre-processing Dataset, Multi-layer Perceptron using TF and TF-IDF features
Debanjan Chatterjee	Pre-processing, Under-sampling, Feature Representation, XGBoost classifier, Passive Aggressive classifier
Gautam Chauhan	dataset modification, pre-processing, plots, bi-LSTM model(with and without network feature), siamese model(with and without network feature)
Shilpa Chatterjee	Pre-processing Dataset, Deep Ensemble Model(CNN+LSTM)[Using GloVe Embedding]

9 Conclusion

The impact of online reviews on society has grown significantly over the past few years. Reviews are crucial to determine success in online businesses. Unfortunately, some users use unfair means to improve their online reputation by writing fake reviews of their businesses or competitors. Hence the need for designing systems that can detect fake reviews is paramount. In the mini-research project we have explored several models which have shown appreciable performance in fake news detection, and tried to create fake review detecting models based on them. The dataset used had high class imbalance in favour of the positive (real review) class, thus under-sampling was performed. Models trained with the under-sampled dataset showed a significant increase in the macro F1 score, but a decrease in accuracy. The macro F1 score is a more reliable metric to determine the performance of the models, hence the trade-off is worth it. From the experimental results, it is observed that models like Passive Aggressive Classifier, Extreme Gradient Boosting, Multinomial Naive Bayes, Multi-layer perceptron and Ensemble model (CNN with LSTM) provide us with the best accuracy and macro F1 score. Additionally, network features were used in a limited way here and with few models. One of the future work is to broaden the network features domain by using other reviews by the user and other reviews by the same user instead of just using ratings.

References

[1] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," 2018.

[2] L. Borges, B. Martins, and P. Calado, "Combining similarity features and deep representation learning for stance detection in the context of checking fake news," *Journal of Data and Information Quality (JDIQ)*, vol. 11, no. 3, pp. 1–26, 2019.

[3] T. Santosh, S. Bansal, and A. Saha, "Can siamese networks help in stance detection?," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, pp. 306–309, 2019.

[4] A. Roy, K. Basak, A. Ekbal, and P. Bhattacharyya, "A deep ensemble framework for fake news-detection and classification," 2018.

[5] H. Wang, Y. Ma, Y. Deng, and Y. Wang, "Fake news detection algorithms comparison and application of xgboost, svm, and nb," *World Scientific Research Journal*, vol. 7, no. 1, pp. 323–329, 2021.

[6] S. Gupta and P. Meel, "Fake news detection using passive-aggressive classifier," in *Inventive Communication and Computational Technologies*, pp. 155–164, Springer, 2021.

[7] X. Wu, S. Cheng, and Z. Chai, "Fake news stance detection," 2017.

[8] A. K. Chaudhry, "Stance detection for the fake news challenge : Identifying textual relationships with deep neural nets,"

[9] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, 2015.

[10] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive aggressive algorithms," 2006.

[11] M. Tosik, A. Mallia, and K. Gangopadhyay, "Debunking fake news one feature at a time," 2018.