Classification of Fake News: A Comparative Study

EMANUELE SANTORO, ROHIT KAUSHIK, and SARA MERENGO

Fig. 1. Fake news word tag cloud

This research study is focused on the detection of fake news using various NLP classification methods. The data used in this study consists of two datasets - one for training and one for testing. They include a mix of both real and fake news. Both datasets have a near 50% of distribution of real and fake news. We compared several models for this classification task, and used a Stylometry embeddings approach as our baseline model with SVM, which reported an accuracy of 89%. As an alternative, GloVe embeddings were also created and used with an LSTM, which resulted in 99.8% accuracy. A generic SVM with PoS tagging did not work very well, with only 49% accuracy. A BERT classifier was also trained, which resulted in an accuracy of 94.9%.

To conclude, out of all the models compared for fake news classification, LSTM neural networks gave the best classification of real and fake news.

Additional Key Words and Phrases: Datasets, Neural Networks, Fake News, GloVe, Stylometry SVM, LSTM, Bayes, BERT

Authors' address: Emanuele Santoro; Rohit Kaushik; Sara Merengo.

Manuscript submitted to ACM

ACM Reference Format:
Emanuele Santoro, Rohit Kaushik, and Sara Merengo. 2023. Classification of Fake News: A Comparative Study. 1, 1 (July 2023), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

While it is clear that fake news have existed since humans began communicating and sharing information [15], the recent communications, Internet, and social media boom has facilitated the generation and spread of news both real and false. Recognizing the peril posed by this dissemination of fake news is crucial as it has the potential to profoundly impact our society and democratic values [5].

Finding an exact and concise definition of Fake News can be challenging, as different factors need to be taken into consideration. [8] defines Fake News as false but verifiable news composed of false facts based on real ones, drafted in a way to trigger an emotional load and aiming to deceive its readers and influence their opinion through an implicit conclusion.

As AI models advance, it is now commonplace to use these increasingly powerful models to predict the authenticity of an item. These AI models leverage the power of machine learning algorithms, natural language processing and data analytics to examine textual content and identify patterns that indicate untruth or inaccuracy [7]. Although artificial intelligence models have proven to be powerful tools, the current state of the art does not provide a deterministic algorithm to determine a priori whether a news article can be classified as false or reliable, as it is difficult even for humans to assess the truth of news [13].

The task of identifying Fake News can be described as a binary classification problem, where the goal is to classify a given news article into one of two classes: FAKE or REAL. This goal can be achieved by analyzing the content and metadata of news articles, including items such as title, text, author, etc. By examining these components, valuable insights can be gained that help distinguish fake news from real news. The goal of this work is to help identify fake news by using linguistic analysis of different news articles to train different models to a reasonable level of accuracy.

2 RELATED WORK

In this section, we provide a literature review of existing fake news detection solutions. These approaches can be divided into three categories: linguistic-based, social-context-based, and knowledge-based [26].

Linguistic-Based Analysis. This method refers to the analysis and accurate examination of natural language ([20], [6],[4]). Its objective is to extract significant data, such as the language proficiency of the news creator, syntactic structures, understandability, psychological cues, and n-gram patterns [24]. These valuable insights are derived exclusively through linguistic analysis of the news content.

Social-Context-Based. This approach analyzes the spreading patterns and the diffusion on social networks to distinguish misleading substances ([19],[33],[34]). The peculiarity of this investigation is to check both the integrity of news and users' credibility in order to generate a probabilistic graphical model capable of classifying the news.

Knowledge-Based Analysis. This approach checks the already existing human knowledge to estimate the likelihood of news being false, using external sources to check the veracity of the claimed news ([11],[29],[32]). This approach can be really useful when facing a specific category of news [14] where it's easier to identify inconsistencies between

Attribute	Real news	Fake news	All news
text - length	574.287	519.241	549.870
text - average word length	4.955	4.793	4.883
text - variance in word length	7.310	6.934	7.143
text - punctuation ratio	0.028	0.031	0.029
text - uppercase ratio	0.048	0.056	0.052
text - type token ratio	0.558	0.570	0.563
text - Yule's k	98.888	94.648	97.007
title - length	11.052	13.127	11.973
title - average word length	5.445	5.439	5.443
title - variance word length	6.043	6.453	6.225
title - punctuation ratio	0.030	0.025	0.028
title - uppercase ratio	0.106	0.287	0.186

Table 1. Average attributes of real and fake articles

trusted and not-trusted sources. Nevertheless, it has some limitations when applied to predicting the truthfulness of very recent news.

3 DATASET EXPLORATION

The dataset used for this project is the WELFake [31] dataset. It is made up of 72,134 articles, of which 35,028 are labelled real and 37,106 are labelled false, with a split of around 48% real news and 52% fake news. For each article, three columns are included: title, text and label. The dataset was created by merging four popular news datasets (i.e. Kaggle, McIntire, Reuters, BuzzFeed Political) to prevent over-fitting of classifiers and to provide more text data for better ML training.

As part of our dataset exploration, we computed the text features shown in Table 1, which will also be later used for our baseline stylometry approach. Some of the features were computed using 'lexicalrichness' module in Python [28] [27].

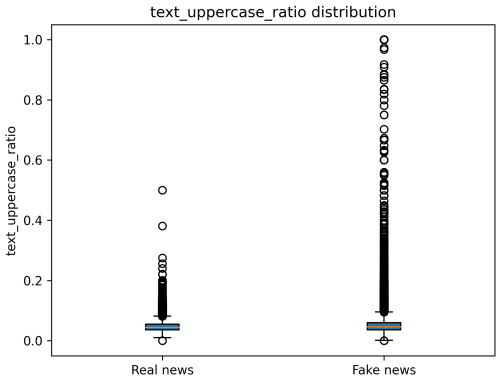
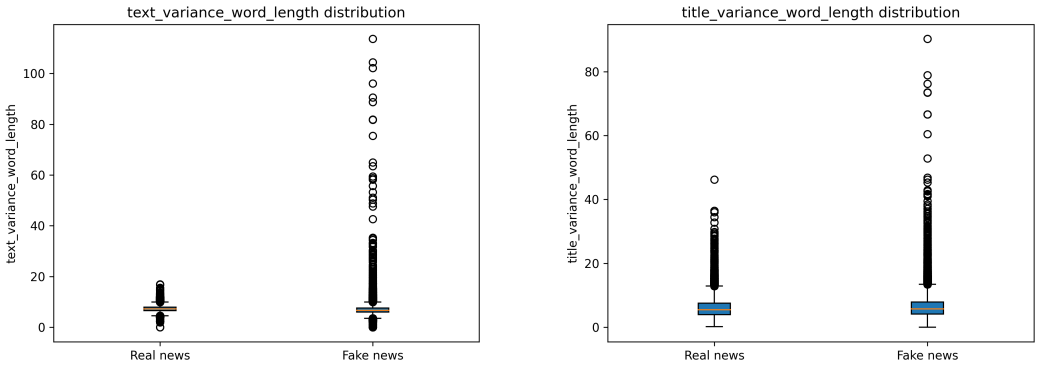


Fig. 2. Difference between text uppercase ratio distribution between real and fake articles



(a) Variance in word length in text

(b) Variance in word length in title

Fig. 3. Difference in variance of word length distribution between real and fake articles

We can see how some features' averages actually show a significant difference between real and fake news, the most notable of which is the title uppercase ratio. We can also observe in Figure 2 how even though the average for the text uppercase ratio is closer for real and fake news, most articles with a high text uppercase ratio fall in the fake news category. These two facts suggest not only that this is valuable information for a stylometry approach, but also that other tools such as GloVe embeddings and BERT models should be case-sensitive.

We can also observe from Figure 3 how even though the average value for the variance of word length is similar between real and fake news, most articles with very high variance values are fake news. By examining these instances we can discover that their high variance in word length is typically due to short texts containing very long links.

We could be surprised by the fact that Yule's k seems to be higher for real news than fake news, as a higher value indicates a smaller vocabulary richness. We can however see from Figure 4 that articles with a high Yule's k tend to be fake news. In any case this particular feature is not always a good indicator as any text without any repeating words scores 0, which makes the score not particularly indicative for short texts.

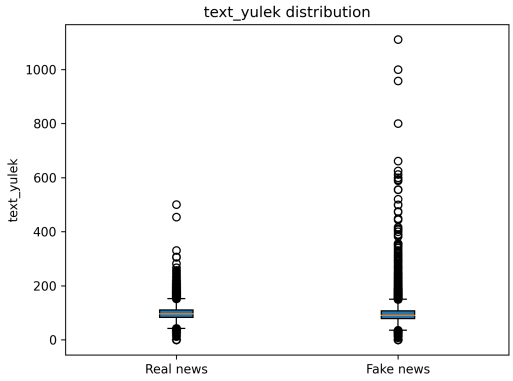


Fig. 4. Difference between text Yule's k distribution between real and fake articles

4 METHODOLOGY

4.1 General Preprocessing

The dataset was acquired as `.csv` file. When working with Python, the framework "pandas" [30] was used to easily manipulate data. A general preprocessing was performed on the whole dataset in order to:

- Remove none values in all columns
- Remove rows that contain no text in the text or title column
- Remove rows in which text or title contain only non-alphanumeric characters
- Remove duplicates

The dataset contained roughly a thousand rows with at least a none value and roughly a thousand rows whose title or text columns were either empty or made up only of non-alphanumeric characters. More significant was the share of duplicate rows, which resulted in removing more than 8,000 rows from the dataset.

The final dataset contains then 62,549 articles of which 34,790 are labelled real and 27,759 are labelled fake, with roughly a 55%/45% split.

4.2 Baseline: stylometry approach

As a first approach, a number of stylometry features (summed up in Table 1) were computed for each article in the dataset. The dataset was then normalized.

An SVM [22] model fit on this data reports an accuracy of 89%, which is a fairly good result considering that this approach is considerably simpler and less expensive in terms of training time and resources than more complex language models. Because of its simplicity compared to other approaches this method was chosen as a baseline for our results.

4.3 Naive Bayes

Naive Bayes models are usually simple probabilistic classifiers, but can still achieve good performance on NLP tasks. They are based on the Bayes Theorem and on the assumption of independence. Each word is treated independently and

a probability is assigned to it. We then calculate the probability of an article being fake for each word it contains and combine these probabilities to obtain the overall probability of the article being fake.

Two methods were used for this approach. In the first approach, we counted the number of occurrences of each word across the entire dataset [1]. In the second approach, we assigned a weight to each word based on TF-IDF (Term Frequency - Inverse Document Frequency) [3]. Table 2 displays the top 10 most common words and the top 10 words with the highest weight.

Word	Occurrences	Weight (TF-IDF)
the	234582	2172.421
said	231350	2395.302
trump	179738	3083.699
would	105027	1231.709
us	101137	1508.095
people	87957	1103.017
one	84484	-
president	82717	1263.974
mr	71979	1084.536
new	70326	1001.078
clinton	-	1332.300

Table 2. Occurrences and weights (TF-IDF) of the most common words

The previous numbers were utilized as input for training a Naive Bayes model [2]. The two approaches were tried both including and not including the title when counting occurrences or calculating weights. It was concluded however that incorporating the title did not significantly alter the results after training the model.
As we can see from the result in Table 3, using the number of occurrences leads to a better result than assigning weight to each word, and the overall accuracy 90% is a good result for this simple approach.

Classification Report						
Approach	**TF-IDF**			**Count Occurences**		
	Accuracy	**Precision**	**F1-Score**	**Accuracy**	**Precision**	**F1-Score**
With Title	0.87	0.87	0.87	0.90	0.91	0.90
Without Title	0.87	0.87	0.87	0.90	0.91	0.90

Table 3. Classification Reports and Accuracy Scores

4.4 GloVe

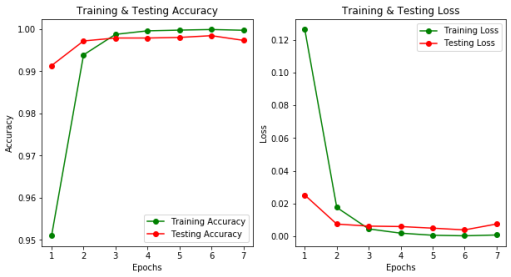
Following the preprocessing of the datasets, a GloVe model [23] was trained based on a document-term matrix created from the ‘text2vec‘ [12] package in R. The document-term matrix represents the frequency or occurrence of each word in each document. The GloVe model was trained using the matrix, to create a model that would be used to run the LSTM algorithm for classifying real and fake news.

4.5 Support Vector Machine Classifier

After loading in the preprocessed dataset, some additional processing was done, such as tokenizing each word in the ‘text‘ column of the data.

	precision	recall	f1-score
False	0.99957	0.99678	0.99817
True	0.99654	0.99954	0.99803
accuracy			0.99811
macro avg	0.99805	0.99816	0.99810
weighted avg	0.99811	0.99811	0.99811

(a) Scores



(b) Accuracy and Loss

Fig. 5. LSTM Performance

The tagger is initialized with a default value of wn.NOUN (noun) from the WordNet corpus in the NLTK library [9]. This means that if the PoS tag is not explicitly specified in the tagger, the default PoS tag assumed will be a noun.

The tagger is then updated with specific mappings for other parts of speech. For example, 'J' is mapped to wn.ADJ (adjective), 'V' is mapped to wn.VERB (verb), and 'R' is mapped to wn.ADV (adverb). This provides more accurate lemmatization for these specific parts of speech. The word is then lemmatized using the WordNetLemmatizer() from the NLTK library. The lemmatization is performed based on the PoS tag, which is obtained from the tagger.

As a result, the SVM [16] model created by this method performed in a subpar manner, and correctly only classified 49.615% of articles correctly.

4.6 LSTM Classifier

The classifier first reads the training set and converts it into a dataframe for further analysis. The columns 'title' and 'text' are concatenated to form a single column with all the string content of the article.

Next, GloVe word embeddings are created from the strings in the articles. The strings are tokenized using 'Keras' [17] and then the tokenized sequences are converted to padded sequences of the same length.

Then, the GloVe model is introduced and converted to a dictionary where each word is mapped to its vector. We then create a word index dictionary from the tokenizer's word index[10].

Finally, we build the model using a 'Sequential' model from Keras. It adds an embedding layer initialized with the GloVe embeddings, followed by an LSTM layer, and two fully connected layers with dropout. The output layer uses a sigmoid activation function since the problem is binary classification. The model is compiled with the binary cross-entropy loss function, Adam optimizer, and accuracy as the metric.

We observe that the LSTM classifier with GloVe embeddings works quite well for binary classification of text, with accuracy of 99.8% and extremely low training loss, as seen in Fig. 4.

4.7 BERT approaches

For BERT-based approaches, the dataset was preprocessed by removing non-alphanumeric characters as well as stopwords, and the title and text fields were concatenated in a single field.

Two different BERT-based models were used:

(1) Distilbert [25] is a smaller, lighter version of BERT, which has 40% less parameters and runs 60% faster while mantaining 95% of BERT's performances as measured on the GLUE language understanding benchmark. The "cased" version (which is case-sensitive) was used with a maximum input length of 32 tokens, reaching an accuracy of 94.9%. Using the cased model provided better results than the uncased version while increasing input length did not significantly improve performance.

(2) Albert [18] is also a smaller version of BERT (the base version has 12 million parameters compared to the 108 million of BERT base) which mantains similar performances to BERT in the base version (which is the one used here) and even outperforms BERT in its xxlarge version, despite having less parameters than BERT large. For this task we used Albert base version 2, which is uncased, and trained it with a maximum input length of 32 tokens, reaching an accuracy of 93.5%. Also in this case the accuracy did not significantly improve when increasing the maximum input length.

4.8 GPT2 approach

Similar to the BERT approach, the dataset was preprocessed by removing non-alphanumeric characters and stopwords and concatenating the title and text field into a single field.

The GPT-2 model [21] is a highly influential language model developed by OpenAI. Despite having a smaller parameter count compared to other models, GPT-2 has demonstrated impressive performance. The base version of GPT-2 comprises 117 million parameters, allowing it to excel in various language tasks, including text classification. For this task, we trained it with a maximum input length of 100 obtaining an accuracy of 93%. Increasing the maximum length did not yield a significant improvement in accuracy.

4.9 Summary

Finally, we show a summary of all the significant results obtained with the different approaches tried in Table 4.

Method	Accuracy	Precision	Recall	F1 score
Stylometry + SVM	0.895	0.917	0.886	0.89
Naive Bayes	0.904	0.905	0.905	0.905
Distilbert	0.949	0.939	0.961	0.95
Albert	0.935	0.974	0.903	0.937
GPT2	0.932	0.937	0.932	0.932
GloVe + LSTM	0.998	0.998	0.998	0.998

Table 4. Summary of obtained results

5 CONCLUSION

The task of detecting fake news varies in complexity, and it is difficult to generalize to all possible news. Even though our model achieves very good accuracy in predicting fake news, we are far from finding a general solution for news classification. This is largely due to the ever-rapid changes in language, topics and concepts presented in the news themselves.

Global events, interests and language will change, and our model may perform poorly trying to classify an article whose topic, context and language don't match the information in our dataset. Nonetheless, our study still shows that,

with sufficiently large datasets, we can achieve reasonably high accuracy in detecting fake news that is conceptually similar to the fake news in the dataset.

REFERENCES

[1] [n. d.]. scikit-learn: CountVectorizer Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html. Accessed: 2023.

[2] [n. d.]. scikit-learn: MultinomialNB Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html. Accessed: 2023.

[3] [n. d.]. scikit-learn: TfidfVectorizer Documentation. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. Accessed: 2023.

[4] H. Ahmed, I. Traore, and S. Saad. 2017. Detection of online fake news using n-gram analysis and machine learning techniques. In *Proceedings of the International Conference on Intelligent Secure Dependable Systems in Distributed and Cloud Environments*. 127–138. https://link.springer.com/book/10.1007/978-3-319-69155-8

[5] Hunt Allcott and Matthew Gentzkow. 2017. Social Media and Fake News in the 2016 Election. *The Journal of Economic Perspectives* 31, 2 (2017), 211–235. http://www.jstor.org/stable/44235006

[6] T. G. de Almeida. 2019. Liardetector: A linguistic-based approach for identifying fake news. (2019).

[7] Athira A B, S D Madhu Kumar, and Anu Mary Chacko. 2022. Towards Smart Fake News Detection Through Explainable AI. arXiv:arXiv:2207.11490

[8] Nicolas Belloir, Wassila Ouerdane, Oscar Pastor, Émilien Frugier, and Louis-Antoine de Barmon. 2022. A Conceptual Characterization of Fake News: A Positioning Paper. In *16th International Conference on Research Challenges in Information Science (RCIS'22)*. Barcelona, Spain. https://hal.science/hal-03679073

[9] Steven Bird and Edward Loper. 2004. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, Barcelona, Spain, 214–217. https://aclanthology.org/P04-3031

[10] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1657–1668. https://doi.org/10.18653/v1/P17-1152

[11] Nancy J. Conroy, Victoria L. Rubin, and Yimin Chen. 2015. Automatic Deception Detection: Methods for Finding Fake News. In *Proceedings of the 78th ASIST Annual Meeting: Information Science with Impact: Research in and for the Community*, Vol. 52. 1–4.

[12] cran R. [n. d.]. text2vec-R. https://cran.r-project.org/web/packages/text2vec/index.html. Accessed: 2023.

[13] Naila Dharani, Jens Ludwig, and Sendhil Mullainathan. 2023. Can A.I. Stop Fake News? The spread of disinformation illuminates algorithms' unique abilities and shortcomings. *CBR - Artificial Intelligence* (January 18 2023). https://www.chicagobooth.edu/review/can-ai-stop-fake-news

[14] Adrian Groza. 2020. Detecting Fake News for the New Coronavirus by Reasoning on the COVID-19 Ontology. *arXiv preprint arXiv:2004.12330* (2020).

[15] Monika Hanley and Allen Munoriyarwa. 2021. *Fake News*. 157–176. https://doi.org/10.1515/9783110740202-009

[16] Gumwon Hong. 2005. Relation Extraction Using Support Vector Machine. In *Second International Joint Conference on Natural Language Processing: Full Papers*. https://doi.org/10.1007/11562214_33

[17] keras.io. [n. d.]. Keras. https://keras.io/guides/functional_api/. Accessed: 2023.

[18] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *CoRR* abs/1909.11942 (2019). arXiv:1909.11942 http://arxiv.org/abs/1909.11942

[19] Yudong Liu and Yao-Feng Barry Wu. 2018. Early Detection of Fake News on Social Media through Propagation Path Classification with Recurrent and Convolutional Networks. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 1–8.

[20] Shafayat Bin Shabbir Mugdha, Sayeda Muntaha Ferdous, and Ahmed Fahmin. 2020. Evaluating Machine Learning Algorithms for Bengali Fake News Detection. In *International Conference on Computer and Information Technology (ICCIT)*. 1–6.

[21] OpenAI. 2023. GPT-2. https://github.com/openai/gpt-2.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[24] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandre Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104* (2017).

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* abs/1910.01108 (2019).

[26] Noureddine Seddari, Abdelouahid Derhab, Mohamed Belaoued, Waleed Halboob, Jalal Al-Muhtadi, and Abdelghani Bouras. 2022. A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media. *IEEE Access* 10 (2022), 62097–62109. https://doi.org/10.1109/ACCESS.2022.3181184

[27] Lucas Shen. 2021. Measuring Political Media Slant Using Text Data. https://www.lucasshen.com/research/media.pdf

[28] Lucas Shen. 2022. LexicalRichness: A small module to compute textual lexical richness. https://doi.org/10.5281/zenodo.6607007

[29] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36.

[30] The Pandas Development Team. YYYY. *Pandas: Powerful Python Data Analysis Library.* https://pandas.pydata.org/

[31] Pawan Kumar Verma, Prateek Agrawal, and Radu Prodan. Year. WELFake dataset for fake news detection in text data. https://doi.org/10.5281/zenodo.4561252

[32] Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task Definition and Dataset Construction. In *Proceedings of the ACL Workshop on Language Technologies and Computational Social Science*. 18–22.

[33] Suhang Yang, Kai Shu, Shimei Wang, Rui Gu, Fang Wu, and Huan Liu. 2019. Unsupervised Fake News Detection on Social Media: A Generative Approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5644–5651.

[34] Xiaomo Zhou and Reza Zafarani. 2019. Network-Based Fake News Detection: A Pattern-Driven Approach. *ACM SIGKDD Explorations Newsletter* 21, 2 (2019), 48–60.

A INDIVIDUAL CONTRIBUTIONS

Task distribution is shown in Table 5

Team member	Tasks
Emanuele Santoro	Pre-processing, Naive Bayes, GPT2
Rohit Kaushik	Pre-processing, GloVe, SVM Classifier, LSTM Classifier
Sara Merengo	Pre-processing, Stylometry approach, BERT approaches

Table 5. Work distribution between team members