02807 PROJECT REPORT: FAKE NEWS CLASSIFICATION

[redacted]

ABSTRACT

This project addresses fake news classification by contrasting a combination of classical machine learning approaches with state-of-the-art model performance, examining the merits of clustering-based ensemble classifiers, and exploring the benefits of integrating graph-based news sharing patterns into classification models. Evaluating on the Kaggle Fake News and BuzzFeed data sets reveals that our ensemble models, particularly those using DBSCAN clustering, exceed global model performance by achieving balanced accuracies of up to 94%, comparable to the performance of state-of-the-art methods. The study confirms that graph representations bolster classification, with effectiveness growing alongside the graph's complexity. These findings endorse graph-augmented ensemble methods as a potent strategy for enhancing fake news classification.

1. INTRODUCTION

Social media news significantly influence public opinion on important topics like presidential elections and healthcare guidelines [18]. Concurrently, research enforces rising global concerns about the spread of fake news, which proliferates faster and more widely than true news, impacting politics, economy, and society [15]. This trend underscores the need for prompt and accurate identification of news truthfullness on social media, to mitigate potential harms.

Motivated by that need, this project introduces a comprehensive method for detecting fake news - by combining a variety of simple - both unsupervised and supervised - machine learning and data science tools for textual analysis and social network theory. As such, this project seeks to answer the following research questions:

1. Can a combination of classical machine learning methods perform comparably to advanced state-of-the-art fake news classification models?
2. Does the integration of varying complexities of local text representation structures improve a purely text-based fake news classification model?
3. How does including graph-based representations of article sharing patterns affect a fake news classification model's generalization, and does this effect grow as more edges are added, simulating the temporal evolution of the graph?

Methods within course curriculum:
Week 1: TF-IDF, Week 6: K-means clustering, DBSCAN, Week 7: social network modeling, spectral clustering.
Methods beyond course curriculum:
Ensemble models, Random Forest (supervised learning methods), node2vec feature representation, Latent Semantic Analysis.

2. RELATED WORK

Current state-of-the-art methods for fake news classification prominently utilize deep learning models, especially ones based on transformer architectures like BERT [2], GPT [11], and variants thereof, known for advanced feature extraction and excelling in nuanced textual understanding, crucial for distinguishing fake news. Complementing these, recent advancements in Graph Neural Networks (GNNs) leverage social network graphs to enhance fake news classification [10, 16]. This multi-aspect approach, integrating content analysis with social context via GNNs, offers a comprehensive method to discern real from fake news. Prior studies indicate that model performance in fake news classification varies by data set [6]; for instance, the top model on the Kaggle Fake News data set outperforms that on the FakeNewsNet-PolitiFact data set by about 11 percentage points, with accuracies of 96.4% and 85.3%, respectively. Accuracies for the top 10 models in the KFN competition range from 83.6% to 98.6%.

3. DATA SETS

To address the research questions, this study leveraged two data sets: the Kaggle Fake News data set (KFN) [8] and the original version of the BuzzFeed-FakeNewsNet data set (BuzzFeed) [13, 14, 12]. The KFN data was restricted to the training part of the original data set, resulting in a total of 18,285 unique news articles after removing observations with missing values. For each article, the title, textual content and a binary label specifying the authenticity of the article were used. The BuzzFeed data set comprises only 182 news articles with analogous details but also incorporates behavioral data pertaining to the sharing of news among Twitter users.

3.1. Preprocessing the Textual Content

All processed news articles had English stopwords removed[1] and were tokenized and Porter stemmed[2]. Resulting word-

[1] Stopwords registered in Python's NLTK library.
[2] https://www.nltk.org/api/nltk.stem.porter.html

clouds are presented in Figure 7 (Appendix D). Using token-vocabularies specific to each data set, TF-IDF matrices with dimensions of $18,285 \times 124,742$ for KFN and $182 \times 6,985$ for BuzzFeed were generated. These matrices provide interpretable feature bases but also present computational challenges due to their high-dimensionality and sparsity, especially for KFN. To address this, dimensionality reduction was applied using Latent Semantic Analysis (LSA or Truncated SVD) on the ℓ_2-normalized-by-document TF-IDF matrices, resulting in a dense embedding space with a reduced dimensionality of $D = 300$. This approach, common for compactly representing large text corpora [3, 4], required computational resources from DTU HPC for implementation. Computational limits led to pre-processing occuring outside cross-validation folds - using all (future test *and* training) data to create the TF-IDF and LSA matrices. Further considerations on this approach and its potential consequences are remarked upon in Appendix A.1.

3.2. Preprocessing the Relational Content

The BuzzFeed data set, containing Twitter interaction data for its articles, allows for a graph representation where nodes represent articles. An edge was formed between two articles, N_1 and N_2, if at least one Twitter user had shared both. Singleton article pairs were excluded to ensure the graph became a single connected component, with edge weights reflecting the number of users sharing both articles. Basic statistics of these data, related to article sharing, are presented in Table 1, while the resulting network is visualized in Figure 1.

Statistic	Graph attr.	Value
Number of news, N	Nodes	182
Number of users, U	-	15,257
N-U interactions	-	22,779
N-N interactions	Edges	4,772

Table 1: Basic statistics of the BuzzFeed data set and the constructed network seen in Figure 1.

3.3. Initial Data Analysis

To visualize the dense LSA representation, t-SNE was employed for dimensionality reduction [17]. A 3-dimensional t-SNE representation of KFN can be seen in Figure 2 (and for BuzzFeed in Figure 8, Appendix D). Here the contents of the 3 dimensions are visualized in a pairwise fashion. Though no label information is used in constructing the embedding space, fake and real news are relatively separable for KFN, even though the shown marginal distributions suggest that the data manifold resembles a large point cloud.

The rightmost affinity matrix in Figure 3 shows higher cosine similarity between real news pairs compared to real-fake or fake-fake pairs in the KFN data set. This trend is not evident in the adjacent matrix for BuzzFeed. However, Figure 1 reveals 4 clusters - two major ones separated into real and fake news, and two smaller mixed-labelled clusters.

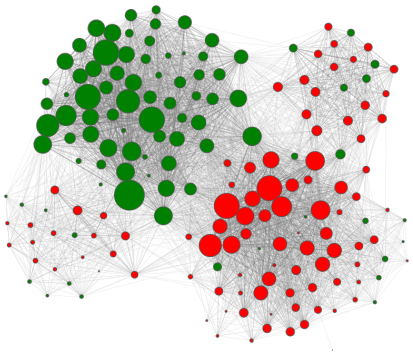


Fig. 1: The BuzzFeed network constructed by linking news articles if at least one Twitter user has both in their sharing history. The two large clusters appear to be related to whether the articles are real (green) or fake (red).

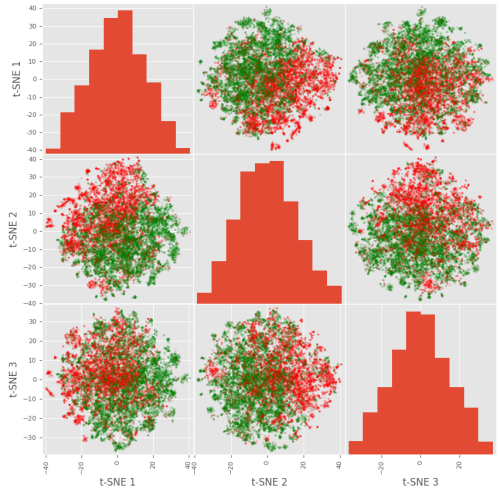


Fig. 2: Dimensionality reduction of the LSA feature space via t-SNE reveals a clear separation between real (green) and fake (red) news articles, suggesting feasibility for the label-free text-encoding classification task.

4. METHODS

This section outlines our methods for fake news classification: text-based Random Forest classification, clustering-based ensembles, graph-based models, bi-modal techniques, graph evolution simulation and evaluation procedures.

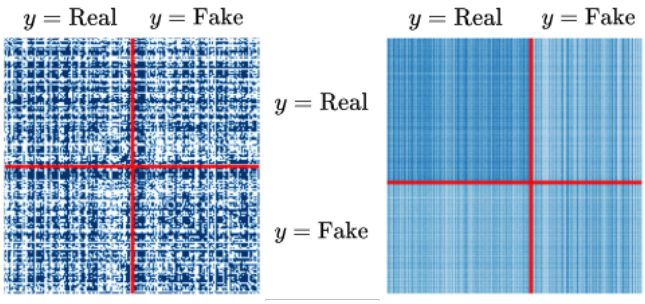


Fig. 3: By-label-ordered affinity matrices showing the cosine similarity between documents within- and between label groups. *Left:* BuzzFeed, *Right:* KFN. A darker blue color reflects a relatively higher cosine similarity.

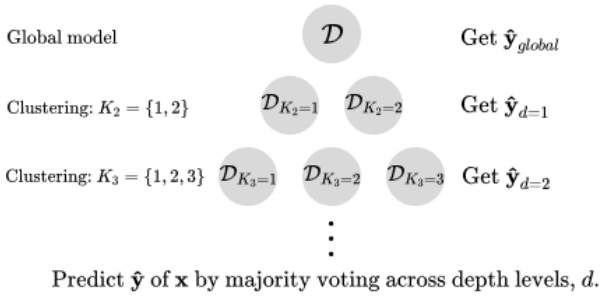


Fig. 4: A simplified overview of the clustering-based ensemble classification approach using K-means as the clustering method. The general idea relies on training a set of locally-aware RF models at each depth level along with a global model. At inference, the input data point, $\boldsymbol{x}$, gets assigned to a cluster at each of the depth levels resulting in a set of depth-dependent predictions, i.e. $\{\hat{y}_0, \hat{y}_1, \cdots, \hat{y}_d\}$ when collecting predicting across locally trained models - i.e. the width - for each depth level. The final prediction of a data point is obtained by regarding each depth level as a member of an ensemble, thereby predicting through majority voting across depth levels.

4.1. Text Based Supervised Learning - Random Forest

Utilizing the data set labels, a supervised learning approach is employed, and given the nonlinear nature of the problem, a RF classification model is chosen. This approach involves training multiple decision tree classifiers on bootstrapped samples, mitigating overfitting through majority voting in the final prediction. The `sklearn` implementation is utilized, employing an ensemble of 100 decision trees for the RF.

4.2. Clustering-based Ensemble Model for Text - Hybrid Unsupervised and Supervised Learning

Inspired by the methods proposed by Li, Huang et al. [7], this project proposes a clustering-based ensemble classification model relying on locally operating models trained on cluster-determined training sets, combined with a global model trained on the full training set, as illustrated in Figure 4. The overall idea is that non-labelled, yet valuable information is structured in the latent LSA feature space. As an example, one could imagine that such local, unstructured patterns would be semantically or syntactically meaningful concepts such as topics. The granularity at which such information is extracted, the *resolution*, is controlled by a depth parameter, d, and a width parameter set W, given by the number of depth levels and a clustering-dependent hyperparameter configuration, respectively. Specifically, the K-means and DBSCAN clustering algorithms are explored for which the W denoted K clusters or a DBSCAN tuple of $(\varepsilon, S_{\min})$, respectively. $S_{\min}$ denotes the minimum number of neighboring datapoints within a Euclidean distance of ε for a point to be considered a core point.

The K-means algorithm, known for its simplicity and efficiency, excels in identifying globular clusters, making it a strong candidate for partitioning our data if such shapes are present. However, its sensitivity to outliers and the requirement of specifying the number of clusters K a priori can be limiting, thus motivating the investigation of the DBSCAN method. In contrast, DBSCAN allows the identification of dense clusters without pre-defining the number of clusters. Furthermore, its robustness to outliers and capability to discover clusters of various shapes and sizes makes it particularly suited for discerning complex patterns in the considered high-dimensional text data. This flexibility is crucial for effectively isolating and analyzing potential misinformation groupings within the data. As DBSCAN does not natively support the assignment of new datapoints and due to the fact that it would be computationally impractical to repeatedly execute the DBSCAN algorithm for each newly introduced test point in a real-time dynamic scenario, we designed a method that assigns these points to existing DBSCAN-derived clusters using the majority cluster label from the k-nearest core points in the training data set. This was implemented with `KNeighboursClassifier` from `sklearn`, setting $k = 10$. Noise points were excluded due to their assumed lack of relevant structural information for local RF models. Refer to Appendix A.2 for further considerations.

4.3. Graph-based RF Model Using Spectral Features

In this project, the proposed formulation by Yang et al. is used [9]. Spectral clustering utilizes an affinity matrix $A \in \mathbb{R}^{N \times N}$, which encapsulates pairwise similarities inherent in the data set - i.e. the adjacency matrix of the graph in this case. Here, the normalized Laplacian matrix is defined as $L = I - D^{1/2} A D^{1/2}$, with D being the diagonal matrix whose (i, i)-element is the sum of A's i-th row. The subsequent clustering task would be carried out using K-means on a basis spectral embedding $V \in \mathbb{R}^{N \times K}$, defined by the eigenvectors associated to the K largest eigenvalues. However, for

the purpose of this project, spectral clustering is solely used to extract a node feature representation basis using the $K = 32$ most prominent eigenvectors of the Laplacian on which a RF model is then fitted. As such, no clustering is carried out.

4.4. Graph-based RF Model Using Node2Vec Features

In the context of node embeddings, the objective is identifying a mapping function $f : V \rightarrow \mathbb{R}^d$ that assigns each node $v \in V$ to a d-dimensional vector, ensuring the preservation of mutual proximity between node pairs in the graph. In *node2vec* [5], the sampling strategy takes the form of a second-order random walk, meaning the transition probability also depends on the previous vertex v_p, ie. $P(v_n|v_c, v_p)$. It does so by applying a bias factor $\alpha_{pq}(v_n, v_p)$ to the edge $(v_c, v_n) \in E$ connecting the current vertex c and a potential next vertex n. This bias factor is defined as a function given by the return parameter p and the in–out parameter q:

$$\alpha_{pq}(v_n, v_p) = \begin{cases} \frac{1}{p}, & \text{if } v_p = v_n \\ 1, & \text{if } v_p \neq v_n \text{ and } (v_n, v_p) \in E \\ \frac{1}{q}, & \text{if } v_p \neq v_n \text{ and } (v_n, v_p) \notin E \end{cases} \quad (1)$$

This approach allows the algorithm to strike a balance between capturing the local and global structure of the graph, resulting in rich and informative embeddings. Similarly to the spectral feature basis, a dimensionality of 32 was chosen.

4.5. Bi-Modal Ensemble Classification

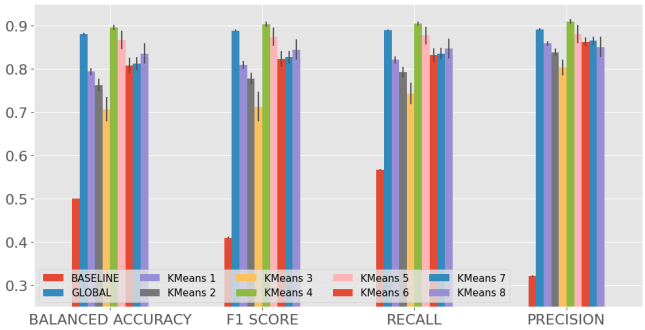
This multi-modal, ensemble-based model combines text and graph-based RF predictions by thresholding the average of their class prediction certainties, $p_{\mathcal{G}}$ and $p_{\mathcal{T}}$, respectively. Thus we let: $\hat{y}_{\text{ensemble}} = \frac{1}{2} \sum_{d \in \{\mathcal{G}, \mathcal{T}\}} p_d \left(y = 1 \mid \boldsymbol{x}\right) \geq \tau$, where $\tau = 0.5$ is the threshold for prediction confidence and $y = 1$ is labeling an article as fake. To simulate dynamic, online sharing of articles and thus the temporal evolution of the resulting graph, we created subgraphs from the original network by progressively removing a set number of random edges, which allowed us to observe the effects of network size on model performance.

4.6. Evaluating Model Performance

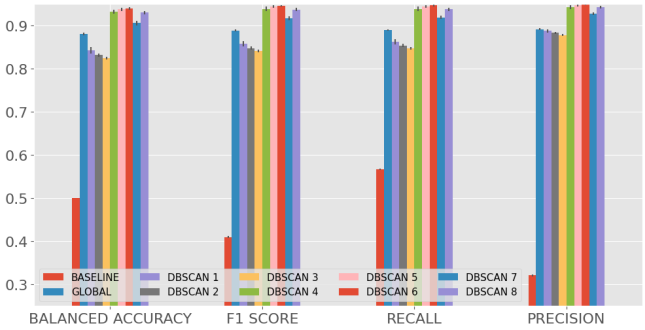
Each algorithm's performance was evaluated using 5-fold Cross-Validation (CV) to estimate generalization error. We reported the average balanced accuracy, weighted F1 score, precision, and recall, along with their standard errors (SEM). For baseline comparison, a majority-voting model was used, predicting the most frequent label in the training data. Hyperparameter optimization was conducted via grid search, with chosen configurations detailed in Appendix B. In this case nested CV was not performed because of computational constraints, which might optimistically bias our estimates of model performances. Further explanations of this modeling choice and its implications are included in Appendix A.3.

5. RESULTS

Figure 5 shows the performance of the clustering-based ensemble model on the KFN data set based on LSA representations with varying hyperparameter configurations. The legend elements, e.g. "KMeans 1", denote specific hyperparameter configurations further detailed in Appendix B. While only a single K-means hyperparameter configuration (i.e. KMeans 4) lead to an ensemble model outperforming the global model (wrt. accuracy), several DBSCAN configurations (DBSCAN 4-8) achieve this.



(a) K-means-based ensemble classification results.



(b) DBSCAN-based ensemble classification results.

Fig. 5: Performances of the clustering-based ensemble classifier on the KFN data set with 5-fold CV for various hyperparameter configurations. Naming conventions and tabular data are found in Appendix B-C.

Figure 6 illustrates the balanced accuracy trends of the graph-based and bi-modal ensemble classifiers on the simulated evolution of the BuzzFeed network in comparison with the strictly text-based model. The graph-based classifiers demonstrate enhanced balanced accuracy with the incremental addition of edges, significantly surpassing the global LSA-based RF classifier at the 30% edge inclusion threshold. Table 2 provides additional metrical results from this experiment utilizing the complete BuzzFeed data set, with both graph-based and LSA representations included. The latter results align with the trend of Figure 6, with all graph-based classifiers significantly outperforming the LSA-based classifier. Also, table 2 elicits no distinguishable performance gap between ensemble models and their purely graph-based counterparts.

Table 2: Classification results for the BuzzFeed data set for a variety of modeling approaches.

	Baseline	**LSA** (global)	**Spectral** (global)	**Spectral** (ensemble)	**node2vec** (global)	**node2vec** (ensemble)
Accuracy	0.500 ± 0.000	0.585 ± 0.041	0.712 ± 0.017	0.728 ± 0.005	0.748 ± 0.018	$\mathbf{0.775 \pm 0.023}$
F1 score	0.298 ± 0.015	0.582 ± 0.041	0.702 ± 0.016	0.720 ± 0.007	0.744 ± 0.018	$\mathbf{0.771 \pm 0.024}$
Precision	0.219 ± 0.013	0.591 ± 0.041	0.731 ± 0.022	0.744 ± 0.008	0.754 ± 0.017	$\mathbf{0.784 \pm 0.022}$
Recall	0.467 ± 0.014	0.583 ± 0.041	0.706 ± 0.017	0.722 ± 0.008	0.744 ± 0.018	$\mathbf{0.772 \pm 0.024}$

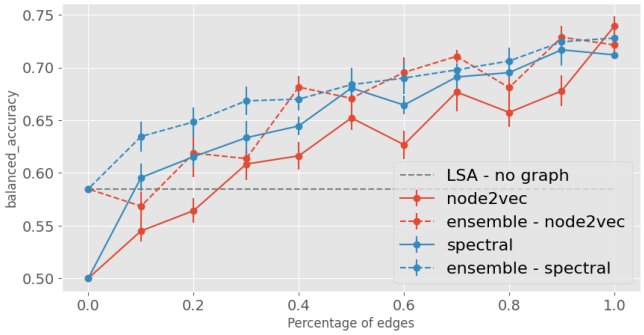


Fig. 6: Performance enhancements for graph and bi-modal ensemble models correlate with increased network connectivity, indicated by the fraction of edges used from the full graph. Figure 9 in Appendix D visualizes the node2vec representation of this network evolution.

6. DISCUSSION

As found from the experiment on the KFN data set, the proposed clustering-based ensemble model for text achieves a maximum performance of 94% balanced accuracy when using DBSCAN with a depth level of $d = 2$ defined by $\varepsilon_{d=1} = 0.75$ and $\varepsilon_{d=2} = 0.5$, respectively with $S_{min} = 3$. This compares to the performance of state-of-the-art methods as presented in Section 2. Though the performance of the best model on the BuzzFeed data is remarkably lower (i.e. 77.5%), this is of no major concern since the solution to the fake news classification problem is highly data set dependent, as previously argued. In this case, the difference in performance is most probably due to a less distinct LSA space (Figure 8, Appendix D) that could be related to the general dynamical structure of both real and fake news. While false positives are undesirable, emphasizing higher precision, the potential harm that can be caused makes achieving high recall crucial in this context. Therefore, we advocate for prioritizing the optimization of recall over precision in future research, despite the presented results indicating marginally higher precision (Figure 5, Table 2, and Appendix C).

The superior performance of the DBSCAN-based model over the global model implies that the LSA feature space contains critical, yet unlabeled, information. This finding supports the notion that integrating auxiliary information, inherently linked to text, can enhance fake news classification. Notably, DBSCAN's adaptability the to data distribution generally gives it an edge over *K*-means. As indicated by Figure 5, the method of incorporating such information—via *K*-means or DBSCAN—has a significant impact, with performance heavily reliant on the chosen hyperparameter configuration.

When considering the relational information contained in the BuzzFeed graph as an additional information source, it turns out that the model performance generally increases, as suggested by Table 2 and Figure 6. Furthermore, this increase appears to be linearly related to the number of edges in the network. When exploiting the full network, there appears to be no significant difference in performance between the bi-modal ensemble models and their purely graph-based counterparts. They all outperform the LSA based model as well. However incorporating both network and text information appears to be a better approach for networks with fewer edges - this holds for spectral features and node2vec. This indicates that relational information and the way in which it is processed is important when discerning fake news.

7. CONCLUSION

This project's investigation into fake news classification revealed the effectiveness of our proposed clustering-based ensemble approach, combining classical, unsupervised, and supervised machine learning methods. Notably, this methodology achieved performance on par with advanced models, particularly on the Kaggle Fake News data set. Key to our success was the strategic use of *K*-means and DBSCAN clustering algorithms, which significantly enhanced our model's capability to discern fake news, hence affirming the value of using clustering techniques to integrate varying complexities of local text representation structures in fake news classification. Additionally, our research showed that incorporating social media sharing history into our models, as tested with the BuzzFeed dataset, significantly boosted performance. This improvement correlated with the graph's growing connectivity, underlining the value of relational data in classifying fake news.

8. REFERENCES

[1] Jason Brownlee. Nested cross-validation for machine learning with python. 2021. URL: https://www.theatlantic.com/product/archive/2014/08/the-hamburger-menu-debate/379145/.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 1810.04805v2, 2019. URL: https://arxiv.org/abs/1810.04805.

[3] Peter Foltz. Latent semantic analysis for text-based research. *Behavior Research Methods*, 28:197–202, 02 1996. doi:10.3758/BF03204765.

[4] Rita González-Márquez, Luca Schmidt, Benjamin M. Schmidt, Philipp Berens, and Dmitry Kobak. The landscape of biomedical research. *bioRxiv*, 2023. URL: https://www.biorxiv.org/content/early/2023/10/16/2023.04.10.536208, arXiv:https://www.biorxiv.org/content/early/2023/10/16/2023.04.10.536208.full.pdf, doi:10.1101/2023.04.10.536208.

[5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016. arXiv:1607.00653.

[6] Matthew Iceland. How good are sota fake news detectors, 2023. arXiv:2308.02727.

[7] Xiaojin Li, Yan Huang, Samden D. Lhatoo, Shiqiang Tao, Laura Vilella Bertran, Guo-Qiang Zhang, and Licong Cui. A hybrid unsupervised and supervised learning approach for postictal generalized eeg suppression detection. *Frontiers in Neuroinformatics*, 16, 2022. URL: https://www.frontiersin.org/articles/10.3389/fninf.2022.1040084, doi:10.3389/fninf.2022.1040084.

[8] William Lifferth. Fake news, 2018. URL: https://kaggle.com/competitions/fake-news.

[9] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm, 2001. URL: https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.

[10] Huyen Trang Phan, Ngoc Thanh Nguyen, and Dosam Hwang. Fake news detection: A survey of graph neural network methods. *Applied Soft Computing*, 139:110235, 2023. URL: https://www.sciencedirect.com/science/article/pii/S1568494623002533, doi:10.1016/j.asoc.2023.110235.

[11] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL: https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf.

[12] Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.

[13] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[14] Kai Shu, Suhang Wang, and Huan Liu. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 2017.

[15] Soroush Vosoughi Sinan Aral, Deb Roy. The spread of true and false news online. *Science*, 2018. URL: https://www.science.org/doi/10.1126/science.aap9559.

[16] Xing Su, Jian Yang, Jia Wu, and Zitai Qiu. Hy-defake: Hypergraph neural networks for detecting fake news in online social networks, 2023. arXiv:2309.02692.

[17] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[18] Mason Walker and Katerina Eva Matsa. News consumption across social media in 2021. *Pew Research Center*, 2021. URL: https://www.pewresearch.org/journalism/2021/09/20/news-consumption-across-social-media-in-2021/.

Appendices

A. FURTHER REMARKS ON SELECTED MODELING APPROACHES

This section highlights relevant considerations related to parts of the processing and modeling pipelines.

A.1. Remark on text preprocessing

Due to memory and computational constraints, our text-encoding pipeline uses all available data points, i.e. what is considered training *and* test data, which introduces a potential bias in our experiments as the model is imbued with additional structural information from the data. Nonetheless, we argue that that in a real-world scenario, this is a non-issue as long as the preprocessing and model are re-calibrated each time new data is acquired. This approach is advocated in practical terms because the nature of fake news is dynamic. For instance, the textual content of a fake news article composed in 2023 is likely to differ significantly from that of an article written in 2016. This underscores the significance of consistently retraining the entire modeling pipeline, including preprocessing and the machine learning model, to stay attuned to evolving trends over time.

A.2. Remark on DBSCAN

When determining the cluster label of a test point, $\hat{c}$ with the DBSCAN algorithm, it could also have been advantageous to start by checking whether it is located within an ε-distance of any core points and if so, label it accordingly. This is is because it resembles how DBSCAN actually operates. However, the approach was deemed computational infeasible for the large KFN data set; both in terms of a high memory print for clusters with many core points initially and additionally due to long running times when countering the memory issue by exploiting mini-batching.

It might also be interesting to consider whether any non-core point would become a new core point after adding the new data points, thereby affecting the cluster assignment. However. this was also deemed computationally infeasible.

A.3. Remark on Hyperparameter selection and Cross-Validation

Recall that when the same CV loop and test data set are used to both select hyperparameters and assess the performance of a model it might lead to an optimistically biased evaluation of model performance [11]. In order to avoid this it is common to use nested cross-validation with an outer and inner CV loop. In this case, the inner loop is used for hyperparameter selection while the outer loop assesses the performance of the models.

Nevertheless, we considered this approach computationally infeasible, given the substantial time required to execute the existing model evaluation loops. Consequently, the results should be interpreted as rougher estimates of the anticipated model performances relative to each other, rather than an accurate reflection of their potential performance in a practical setting.

B. HYPERPARAMETER OPTIMIZATION

For the clustering-based ensemble model, grid search is performed over width- and depth-related parameters with the depth ranging from 1 and 4 depth levels. For the *K*-means-based model the widths considered were $K \in \{2, 4, 6, 8\}$ clusters per layer, where the DBSCAN-based model specified the width through the ε parameter, such that $\varepsilon \in \{0.8, 0.75, 0.675, 0.6, 0.5\}$, where $S_{\min} = 3$ and Euclidean distance was used as the metric. The ε-values were determined based on pilot runs and yielded $\approx$ 2-16 clusters depending on the specific value.

Recall that d specifies the depth level. Here, $K_d = [w_{d=1}, w_{d=2}, \cdots w_{d=d}]$ refers to the number of clusters, i.e. the width, w, of the corresponding depth level. Similarly for DBSCAN, ε_d refers to width-determining parameter at the corresponding depth level, d. Further recall that $S_{\min} = 3$ and the metric used is Euclidean distance.

K-means $1 : d = 1, K_d = [2]$	DBSCAN $1 : d = 1, \varepsilon_{d=1} = 0.8$
K-means $2 : d = 1, K_d = [4]$	DBSCAN $2 : d = 1, \varepsilon_{d=1} = 0.75$
K-means $3 : d = 1, K_d = [6]$	DBSCAN $3 : d = 1, \varepsilon_{d=1} = 0.5$
K-means $4 : d = 2, K_d = [2, 4]$	DBSCAN $4 : d = 2, \varepsilon_{d=1} = 0.8, \varepsilon_{d=2} = 0.75$
K-means $5 : d = 2, K_d = [2, 6]$	DBSCAN $5 : d = 2, \varepsilon_{d=1} = 0.8, \varepsilon_{d=2} = 0.5$
K-means $6 : d = 3, K_d = [2, 4, 6]$	DBSCAN $6 : d = 2, \varepsilon_{d=1} = 0.75, \varepsilon_{d=2} = 0.5$
K-means $7 : d = 3, K_d = [2, 4, 8]$	DBSCAN $7 : d = 3, \varepsilon_{d=1} = 0.8, \varepsilon_{d=2} = 0.75, \varepsilon_{d=3} = 0.5$
K-means $8 : d = 4, K_d = [2, 4, 6, 8]$	DBSCAN $8 : d = 5, \varepsilon_{d=1} = 0.8, \varepsilon_{d=2} = 0.75, \varepsilon_{d=3} = 0.675, \varepsilon_{d=4} = 0.6, \varepsilon_{d=5} = 0.5$

C. NUMERICAL RESULTS RELATED TO FIGURES

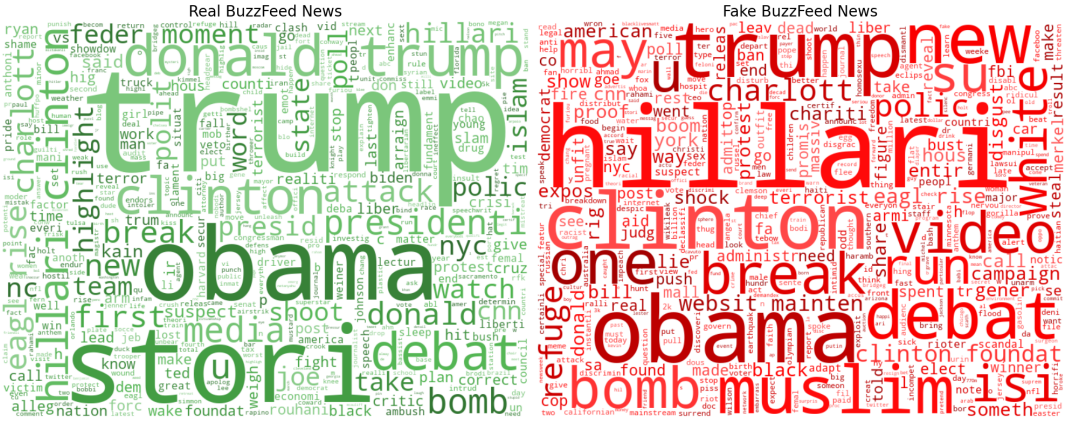
Table 3: Data related to Figure 5a on *K*-means optimization results.

Method	Accuracy	F1 score	Precision	Recall
Baseline	0.500 ± 0.000	0.410 ± 0.002	0.321 ± 0.002	0.567 ± 0.002
LSA	0.881 ± 0.003	0.889 ± 0.002	0.892 ± 0.002	0.890 ± 0.002
K-means 1	0.794 ± 0.008	0.810 ± 0.009	0.859 ± 0.005	0.821 ± 0.008
K-means 2	0.763 ± 0.014	0.778 ± 0.014	0.839 ± 0.008	0.793 ± 0.012
K-means 3	0.707 ± 0.028	0.713 ± 0.034	0.804 ± 0.018	0.743 ± 0.025
K-means 4	0.896 ± 0.006	0.904 ± 0.006	0.910 ± 0.005	0.905 ± 0.005
K-means 5	0.867 ± 0.021	0.875 ± 0.021	0.881 ± 0.021	0.877 ± 0.021
K-means 6	0.808 ± 0.018	0.823 ± 0.018	0.863 ± 0.010	0.832 ± 0.016
K-means 7	0.812 ± 0.015	0.827 ± 0.015	0.865 ± 0.010	0.835 ± 0.014
K-means 8	0.836 ± 0.024	0.845 ± 0.024	0.851 ± 0.023	0.847 ± 0.023

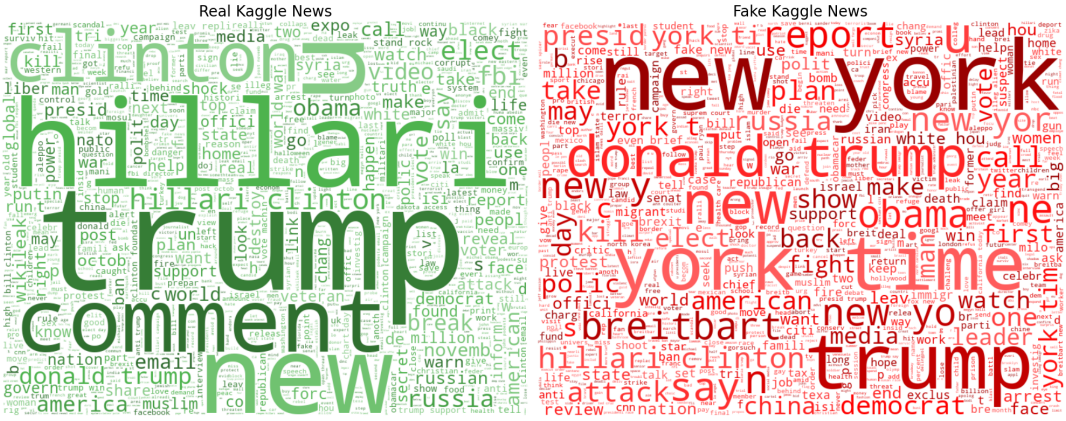
Table 4: Data related to Figure 5b on DBSCAN optimization results.

Method	Accuracy	F1 score	Precision	Recall
Baseline	0.500 ± 0.000	0.410 ± 0.002	0.321 ± 0.002	0.567 ± 0.002
LSA	0.881 ± 0.003	0.889 ± 0.002	0.892 ± 0.002	0.890 ± 0.002
DBSCAN 1	0.843 ± 0.007	0.858 ± 0.006	0.888 ± 0.004	0.863 ± 0.006
DBSCAN 2	0.832 ± 0.004	0.848 ± 0.004	0.883 ± 0.002	0.854 ± 0.004
DBSCAN 3	0.825 ± 0.003	0.841 ± 0.003	0.878 ± 0.002	0.848 ± 0.003
DBSCAN 4	0.932 ± 0.005	0.939 ± 0.005	0.943 ± 0.004	0.939 ± 0.005
DBSCAN 5	0.938 ± 0.004	0.944 ± 0.003	0.948 ± 0.003	0.945 ± 0.003
DBSCAN 6	0.940 ± 0.003	0.946 ± 0.002	0.949 ± 0.002	0.947 ± 0.002
DBSCAN 7	0.906 ± 0.005	0.917 ± 0.004	0.928 ± 0.003	0.919 ± 0.004
DBSCAN 8	0.930 ± 0.004	0.937 ± 0.004	0.943 ± 0.003	0.938 ± 0.004

D. ADDITIONAL VISUALIZATIONS



(a) Kaggle Fake News data set.



(b) BuzzFeed-FakeNewsNet data set.

Fig. 7: Tokenized wordclouds for real (green) and fake (red) news articles. The size of a token is proportional to the term frequency of that token. Though both data sets contain names of political figures there appears to be a difference between KFN and BuzzFeed when considering the content in each category.

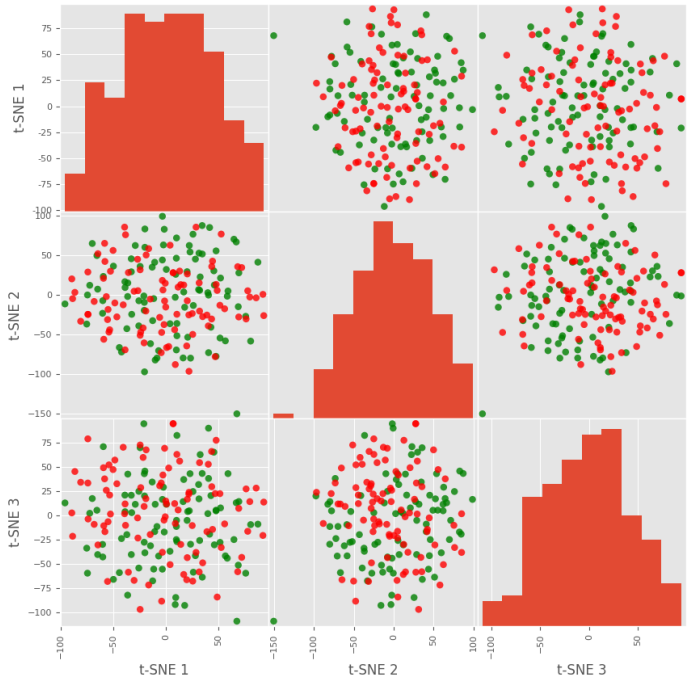


Fig. 8: t-SNE visualization of the LSA features for BuzzFeed. It is quite clear that the Fake/Real news distribution is not as easily visible from the raw data as it is for the KFN data set.

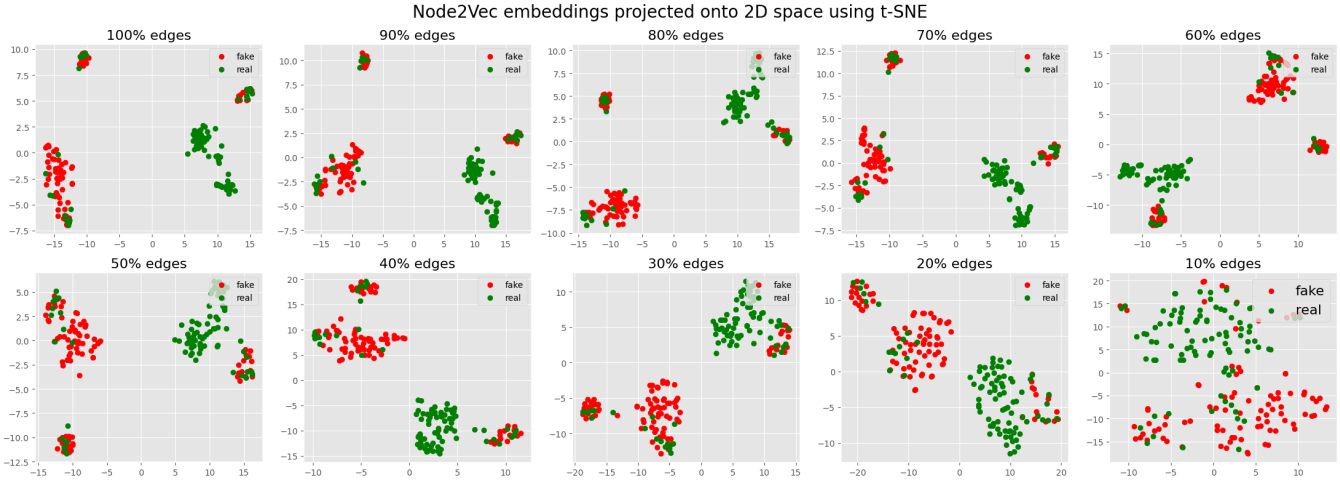


Fig. 9: t-SNE visualiztion of the node2vec features for BuzzFeed for varying fractions of edges in the full graph included.

E. GROUP MEMBER CONTRIBUTIONS

As mandated by the formal requirement, which emphasizes the need for transparency in individual contributions to ensure grading fairness, the group and its members wish to affirm that each member has participated and delivered a commendable and closely equal contribution across the different sections of the project. Should further clarification be required, please feel free to contact any of the group members.