Fake News Detection Using Machine Learning Algorithms

Karsin Dass
University of Michigan
kdass@umich.edu

Maya Ruder
University of Michigan
mayarud@umich.edu

Abstract

The abundance and accessibility of fake news poses a significant challenge to societal and political landscapes exisiting within the digital age, taxing public trust in information and media. This study seeks to explore various machine learning techniques for the detection of fake news by leveraging Naive Bayes, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision Trees, and Random Forest classifiers. We implemented a robust pre-processing pipeline — including tokenization, Term Frequency-Inverse Document Frequency (TF-IDF), word vectorization, and feature selection — to enhance classification accuracy. Our experimental analysis evaluates each model across a diverse range of datasets, evaluating on metrics such as accuracy, precision, recall, and F1-score. This research aims to highlight both the strengths and limitations of machine learning algorithms in text-based classification tasks, offering insights into practical approach for identifying misinformation.

1 Introduction

Fake news has evolved into one of the most pressing threats to achieving an informed society in the digital age, its widespread dissemination disrupting information ecosystems and nurturing an environment of deception and manipulation of the public. This study seeks to construct a pipeline for text classification by implementing machine learning algorithms to effectively detect and classify fake news. By using structured pre-processing steps and experimenting with various classifiers, we can assess the performance of different machine learning models on multiple datasets when applied to the problem of fake news.

2 Related Work

The efficacy of machine learning approaches for fake news detection have been well-documented in recent research. While Naive Bayes models are known for their simple and effective approach to text classification tasks, more advanced techniques like neural networks and combined models tend to offer higher accuracy at the cost of interpretability due to complexity. This study builds on these findings by honing in on the practical application of Naive Bayes and TF-IDF vectorization and examining how other machine learning algorithms perform in a high-context, text-classification setting.

2.1 Logistic Regression

[1] Ahmed H., Traore I., and Saad S. Detection of online fake news using n-gram analysis and machine learning techniques. *Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, 2017, Vancouver, Canada, Springer, pp. 127–138. `https://doi.org/10.1007/978-3-319-69155-8_9`.

Preprint. Under review.

[2] Adeyiga, Johnson Adeleke, et al. "Fake News Detection Using a Logistic Regression Model and Natural Language Processing Techniques." *Bells University of Technology*, Research Article.

2.2 Discriminant Analysis

[3] Kesarwani, Ankit, et al. "Fake News Detection on Social Media using K-Nearest Neighbor Classifier." *2020 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2020, pp. 1-4.

[4] Ogunsuyi, Opeyemi J., and Adebola K. Ojo. "K-Nearest Neighbors Bayesian Approach to False News Detection from Text on Social Media." *I. J. Education and Management Engineering*, vol. 12, no. 4, 2022, pp. 22-32. MECS, 8 Aug. 2022, doi:10.5815/ijeme.2022.04.03.

2.3 Naive Bayes

[5] Z. Khanam et al. 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1099 012040.

[6] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, Ukraine, 2017, pp. 900-903, doi: 10.1109/UKRCON.2017.8100379.

[7] Moorpani, Manisha, et al. "Fake News Detection Using Naïve Bayes Algorithm." *International Research Journal of Modernization in Engineering, Technology and Science*, vol. 4, no. 4, Apr. 2022, www.irjmets.com.

2.4 Neural Networks as Decision Trees

[8] Zhou X., R.J.A., Zafarani C.S. A survey of fake news: Fundamental theories, detection methods, and opportunities. 2020;53(5):1–40.

[9] S. Patil, S. Vairagade, and D. Theng, "Machine Learning Techniques for the Classification of Fake News," 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA), Nagpur, India, 2021, pp. 1-5, doi: 10.1109/ICCICA52458.2021.9697267.

[10] Wang W. Y., *Liar, Liar Pants on Fire: A New Benchmark Dataset for Fake News Detection*, 2017, Association for Computational Linguistics, Stroudsburg, PA, USA.

[11] Islam, M.R., Liu, S., Wang, X., et al. Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, vol. 10, 82, 2020. https://doi.org/10.1007/s13278-020-00696-x.

[12] Fayaz, M., Khan, A., Bilal, M., et al. Machine learning for fake news classification with optimal feature selection. *Soft Computing*, vol. 26, pp. 7763–7771, 2022. https://doi.org/10.1007/s00500-022-06773-x.

[13] Varshney, D., Vishwakarma, D.K. Hoax news-inspector: a real-time prediction of fake news using content resemblance over web search results for authenticating the credibility of news articles. *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 8961–8974, 2021. https://doi.org/10.1007/s12652-020-02698-1.

2.5 Reference Review

The literature surrounding fake news detection has explored various machine learning models, each with its strengths and limitations. Adeyiga et al. found that logistic regression, while interpretable, is often outperformed by more complex models such as random forests, which can capture intricate patterns in text data. Kesarwani et al. demonstrated the potential of distance-based classifiers like K-Nearest Neighbors (KNN), achieving over 90% accuracy, showing its promise for this task. Granik and Mesyura highlighted the effectiveness of Naive Bayes, particularly when used with text vectorization techniques such as Term Frequency-Inverse Document Frequency (TF-IDF). Their work was further expanded by Moorpani et al., who compared different types of Naive Bayes models, identifying Bernoulli Naive Bayes as the best performer for binary classification tasks. Finally, Islam et al. reviewed the use of deep learning techniques and showed that neural networks, especially deep architectures, outperformed traditional models by capturing more complex patterns in textual data. Together, these studies provide a solid foundation for developing fake news detection classifiers, emphasizing the importance of choosing the right model based on the trade-off between interpretability, accuracy, precision, and computational efficiency.

When comparing the findings of previous work to this study, the efficacy of Naive Bayes in a Natural Language Processing setting is affirmed. Interestingly, this study contradicts the results from the Kesarwani et al. study, our findings emphasizing the weakness of distance-based classifiers such as K-Nearest Neighbors in a high-dimensional space. KNN suffers from the curse of dimensionality: a well-documented statistical phenomenon in which the Euclidean distance between data points loses statistical meaning and thus negatively impacting its classification accuracy.

3 Methodology

3.1 Datasets and Pre-processing

Datasets:

- **Albanian Dataset:** Articles, pre-classified as fake or real news, including linguistic and structural features in Albanian. This dataset has 3994 entries.
- **Soccer Dataset:** Social media posts — or "tweets" — related to soccer, labeled as fake or real. Titles and metadata (author, data created, date modified, etc.) from articles included. This dataset has 41868 entries.

Pre-processing:

- Creation of binary column taking values 1 (real news) and 0 (fake news).
- Conversion of text to lowercase strings and removal of NaN values.
- Tokenization to represent text in terms of words and punctuation.
- Experimental implementation of word2Vec embeddings to compute averages of vectorized documents as features.
- Split the data into 80% training and 20% testing.
- Application of TF-IDF vectorization for KNN, SVM, Decision Trees, and Random Forests.

3.2 Machine Learning Models

- **Naive Bayes (NB):** Implements Laplace smoothing and uses log probabilities to make predictions, utilizing both word count and TF-IDF representations.
- **K-Nearest Neighbors (KNN):** Classifies samples based on proximity (in Euclidean distance) to their k nearest neighbors in the feature space. Optimized for small datasets due to computational intensity on larger datasets (curse of dimensionality).
- **Support Vector Machines (SVM):** Maximizes the distance between classes separated in a high-dimensional space by a constructed hyperplane. Features are standardized to scale and represented by TF-IDF application.
- **Decision Trees (DT):** Optimizes interpretable classification rules, separated by splits.
- **Random Forest (RF):** Maximizes predictive power by combining predictions from multiple decision trees trained on random subsets of data and extracted features.
- **Linear Discriminant Analysis (LDA):** Assuming equal covariance among classes, LDA projects data into a lower-dimensional space. LDA has much difficulty handling large datasets, and does not perform effectively in a classification task. **Discarded** from further analysis.
- **Quadratic Discriminant Analysis (QDA):** Builds on LDA by allowing classes to have variation in their covariance matrices. While QDA provides more flexibility, it had a similar problem in usability as LDA. **Discarded** from further analysis.
- **Logistic Regression (LR):** Assuming a linear decision boundary, LR models the probability of binary outcomes using TF-IDF features. LR suffers from over-simplicity, and is consistently outperformed by more complex models. **Discarded** from further analysis.
- **Neural Networks (NN):** Use multi-layer perceptron (MLP) for decision boundaries of increased complexity. Suffers from excess complexity and lack of interpretability. **Discarded** from further analysis.
- **Hybrid Models:**
  - **Naive Bayes + KNN:** Attempts to balance the probabilistic strength of Naive Bayes with the distance-based method of KNN.
  - **Naive Bayes + Logistic Regression:** Attempts to improve upon Naive Bayes by combining probabilistic strength with discriminative power.

4 Experiments and Results

The datasets were split into 80% training and 20% testing. Each model was then evaluated on accuracy, precision, recall, and F1-score. Table 1 summarizes the results for each classifier.

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Naive Bayes	Albanian	0.96	0.95	0.97	0.96
Naive Bayes	Soccer	0.94	0.93	0.94	0.94
KNN	Albanian	0.72	0.82	0.71	0.76
KNN	Soccer	0.60	0.78	0.62	0.69
SVM	Albanian	0.69	0.70	0.72	0.71
SVM	Soccer	0.91*	0.90	0.92	0.91
Decision Tree	Albanian	0.85	0.83	0.86	0.84
Decision Tree	Soccer	0.85*	0.84	0.86	0.85
Random Forest	Albanian	0.87	0.86	0.88	0.87
Random Forest	Soccer	0.91*	0.91	0.92	0.91
Logistic Regression	Albanian	0.87	0.88	0.86	0.87
Logistic Regression	Soccer	0.92	0.92	0.93	0.92
Hybrid KNN/Naive Bayes	Albanian	0.85	0.87	0.83	0.85
Hybrid KNN/Naive Bayes	Soccer	0.94	0.93	0.94	0.94
Hybrid Naive Bayes/LR	Albanian	0.89	0.89	0.90	0.89
Hybrid Naive Bayes/LR	Soccer	0.94	0.93	0.94	0.94

Table 1: Performance Metrics for Classifiers on Albanian and Soccer Datasets.

Note: Models for the Soccer dataset with an asterisk (*) were evaluated on a subset of the data to minimize compile time.

4.1 Feature Analysis

The feature importance was analyzed for each model using different methodologies to extract influential features.

Naive Bayes Naive Bayes calculates the posterior probability $P(C|X)$ for a given class C and feature vector X using Bayes' Theorem (assuming independence of features):

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}.$$

Since $P(X)$ is constant for all C, we maximize the numerator:

$$P(C|X) \propto P(C) \prod_{i=1}^{n} P(x_i|C),$$

where $P(x_i|C)$ is the likelihood of feature x_i given class C.

The influence of an individual word is evaluated by their contribution to the likelihood ratio:

$$\log \frac{P(C_1|X)}{P(C_2|X)} = \log \frac{P(C_1)}{P(C_2)} + \sum_{i=1}^{n} \left(\log P(x_i|C_1) - \log P(x_i|C_2) \right).$$

The log-odds values reflect the significance of individual words in predicting the validity of a news document, in which higher values indicate a stronger "pull" to a "fake news" classification. The top 10 words that "pulled" toward fake news for the Albanian dataset are listed as follows:

Word	Log-Odds Pull
shtype	6.9903
reklamen	6.7493
marketingun	6.2870
pasi	6.1973
sekonda	6.1929
keni	6.1929
hapur	6.1929
posht	5.7624
pamje	5.7485
ketij	5.5669

Table 2: Top Words Pulling Toward Fake News (Albanian Dataset)

For the Soccer dataset, similar analysis revealed the following influential words:

Word	Log-Odds Pull
https	7.9729
co	7.9726
ontime_news	7.9094
scheduled	5.5415
preparations	5.2562
witnessed	5.0366
counterpart	5.0066
ontimenews	4.9756
goalless	4.9173
josevaldo	4.9073

Table 3: Top Words Pulling Toward Fake News (Soccer Dataset)

K-Nearest Neighbors (KNN) KNN classifies a data point X by classifying among its k-nearest neighbors in the feature space, using the distance metric:

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2},$$

TF-IDF (Term Frequency-Inverse Document Frequency) scores numerically represent the extracted text features:

$$\text{TF-IDF}(t, d) = \frac{\text{count of } t \text{ in } d}{\text{total terms in } d} \cdot log(\frac{N}{1 + \text{number of documents containing } t)}$$

The features with the top TF-IDF scores are displayed below — these are the words most influential in fake news classification. When translated, these features reveal themselves to be exceedingly common words, highlighting the weaknesses and blind spots of KNN in a high-dimensional space in which distance between points becomes less meaningful.

Feature	TF-IDF Score
të	0.1393
në	0.0832
dhe	0.0535
për	0.0509
ka	0.0462
me	0.0457
që	0.0428
një	0.0402
se	0.0392
është	0.0337

Table 4: Top Features Influencing Classification (KNN - Albanian Dataset)

For the Soccer dataset:

Feature	TF-IDF Score
the	0.1099
of	0.0532
al	0.0446
and	0.0444
in	0.0423
to	0.0373
is	0.0341
ahly	0.0299
for	0.0264
team	0.0253

Table 5: Top Features Influencing Classification (KNN - Soccer Dataset)

Support Vector Machines (SVM) SVM finds the hyperplane $w \cdot x + b = 0$ that maximizes the distance between classes. The optimization problem is:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

The Radial Basis Function (RBF) kernel $K(x, x') = \exp(-\gamma\|x-x'\|^2)$ is implemented for the text classification problem.

Using the TF-IDF vectorized features, SVM achieved high accuracy on both datasets, as well as a precision of 0.9 for the Soccer dataset and an F1-score of 0.89.

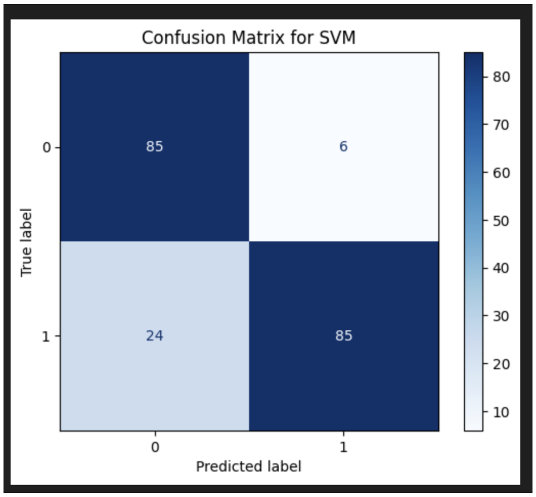

Figure 1: This is the confusion matrix for the trimmed soccer dataset

Decision Tree and Random Forest Decision Trees split data by selecting features that maximize the following equation:

$$IG = -\sum_{c \in C} P(c) \log P(c) - \sum_{i=1}^{k} \frac{|S_i|}{|S|} H(S_i),$$

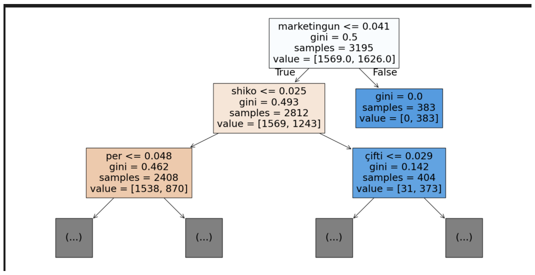

Figure 2: Decision Tree for the Albanian dataset

Random Forest is an ensemble model of Decision Trees. Both models were able to output feature importance but required parameter tuning. Decision Tree ultimately ended up performing best at a maximum depth of 10 with 85% accuracy, while Random Forest performed best at 14 estimators with 87% accuracy.

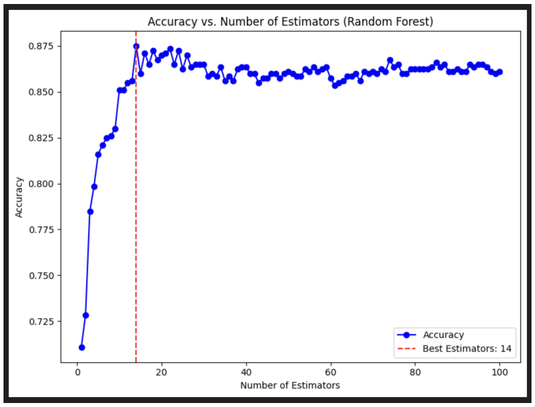

Figure 3: Accuracy based on number of estimators (Albanian dataset)

5 Conclusion and Discussion

By evaluating multiple classifiers — including Naive Bayes, K-Nearest Neighbors, Support Vector Machines, Decision Trees, and Random Forests — this study seeks to demonstrate the challenges of employing machine learning techniques and algorithms in a text classification problem. Throughout our work on this study, we gleaned valuable insights into both the capabilities and limitations of these models in the problem of fake news detection.

5.1 Summary of Key Findings

Naive Bayes is well-documented as an approach that is both computationally efficient and interepretable, excelling in isolating text features, particularly in the Albanian and Soccer datasets, proving its utility in real-world, real-time applications. KNN and SVM were able to achieve a broader contextual understanding of the problem at hand, by implementing TF-IDF vectorization. However, KNN struggled greatly to overcome the curse of dimensionality and SVM exhausted computational resources despite producing high accuracy results. In both instances, the weaknesses of distance-based classification approaches are emphasized. Like SVM, Random Forest struggled to overcome complexity, requiring extensive parameter tuning but producing high accuracy results. Despite high accuracy and precision, the implementation of SVM and Random Forest demand computational resources and parameter optimization that outweighs the accuracy of their models.

While improvements were marginal, ensemble models like combining Naive Bayes with KNN and Naive Bayes with Logistic Regression showed some promise in their ability to leverage the strengths of probabilistic power and discriminative techniques, suggesting room for further exploration in hybrid techniques.

This study also highlights the important of taking appropriate pre-processing steps in classification problems. Construction of a viable pipeline — featuring tokenization, TF-IDF vectorization, feature selection, and so on — played an invaluable role in the performance of each of our models. These steps also allowed us to highlight both the linguistic and contextual differences between real news and fake news, thus improving analysis of results and model interpretability.

5.2 Advantages and Disadvantages

While we should proceed with caution when implementation simpler models like Naive Bayes due to their tendency to oversimplify the complexity of textual data and, in turn, diminish accuracy rates in more nuanced textual situations, Naive Bayes performs exceedingly well in the complex space of detecting fake news. Naive Bayes optimized both speed and ease of implementation, requiring limited computational resources, and deeming itself suitable as a real-world solution. While Random Forest and SVM yielded high accuracy rates, their demands for computational cost and complexity do not make them likely solutions in fake news detection. KNN fails to perform in this task due to a basis in distance metrics, which become less meaningful as the data enters a high-dimensional feature space.

5.3 Future Work and Recommendations

Future work should explore the integration of deep learning techniques to capture the more nuanced and contextual patterns found in textual data, such as the kind used in this study to build a fake news classifier. These more complex models have greater capacity and ability to identify complex linguistic and contextual features that would have otherwise been overlooked by simpler models. In addition to deep-learning-integrated models, it would be worth exploring the capabilities of ensemble models to provide practical solutions to the fake news classification problem. These models would ideally be able to balance enhancing accuracy and ensuring scalability by leveraging the strength of various known methods.

Other areas for improvement would be related to interpretability, computational efficiency, and dataset bias should also be addressed in future work in order to develop practical and scalable solutions that are generalizable across different languages, societies, and contexts.

5.4 Conclusion

This study not only advances practical understanding of machine learning application in text classification problems, but also highlights the critical importance of a multi-faceted approach when seeking to combat misinformation with robust and scalable solutions. We encourage statisticians to build on these findings so machine learning applications may be used to ensure a more informed and protected digital society.