Leveraging Machine Learning for Browser-Based Detection of Misinformation: Towards User-Empowered News Consumption*

Oluwaseun Bukky Afolabi[1], Safina Showkat Ara[2]
Faculty of Technology, Department of Computer Science
University of Sunderland, United Kingdom
afolabi.rob@gmail.com, safina.ara@sunderland.ac.uk

***Abstract*—The surge of fake news on digital platforms presents a pressing societal concern, undermining trust and decision-making processes. The reliability of information, crucial for individuals and societies, faces unprecedented challenges. The rapid evolution of fake news tactics exacerbates this problem, demanding constant adaptation of countermeasures. In response, this study proposes an innovative solution: a user-friendly browser plugin employing machine learning for real-time fake news detection. We conduct a thorough examination of existing techniques, evaluating various algorithms to enhance accuracy. Through rigorous data preparation and algorithm refinement, we achieve significant improvements, emphasizing the importance of textual features and class balancing. The research extends beyond theory with the development and deployment of a practical browser plugin, enabling users to actively combat misinformation. Ethical, legal, and social considerations are integral, ensuring responsible deployment, bias mitigation, and adherence to copyright. The study advocates for ongoing refinement, highlighting the persistent relevance of fake news detection in an information-driven society.**

***Index Terms*—fakenews, machine learning, browser plugin, Misinformation, Infodemic**

I. Introduction

In the digital age, the widespread dissemination of misinformation and fake news presents a significant threat to information integrity, public discourse, and societal stability. Addressing this urgent challenge requires robust mechanisms for detecting and combating fake news effectively. This paper aims to contribute to these efforts through a comprehensive investigation into fake news detection, leveraging advanced machine learning techniques and insights from natural language processing research.

Our primary objective is to develop and evaluate machine learning models capable of accurately identifying fake news articles among the vast online content. By doing so, we aim to enhance the authenticity and reliability of information accessed by individuals worldwide. Our research provides insights into the nature and scope of the fake news problem, reviews pertinent literature, and proposes a methodological approach to address this pervasive issue.

We employ a diverse range of machine learning algorithms, including traditional classifiers and cutting-edge transformer-based models, to explore the efficacy of different methodologies in fake news detection. Through systematic testing of techniques for data preprocessing, feature engineering, and model optimization, we aim to uncover effective strategies for distinguishing between genuine and fabricated news articles.

The principal results of our experiment are expected to provide valuable insights into the performance of different machine learning models in detecting fake news. We aim to identify unique solutions for mitigating the spread of misinformation and enhancing the credibility of online information sources.

In summary, this paper offers a systematic analysis of methodologies, experimental results, and implications for addressing the critical societal challenge of fake news. By advancing our understanding of fake news detection techniques, we contribute to the development of more robust and reliable mechanisms for ensuring information integrity in the digital age. The structure of this research paper includes a literature review in Section 2, dataset and methodology details in Sections 3 and 4, evaluation and analysis of results in Section 5, potential avenues for further research in Section 6, and conclusions in Section 7.

II. Literature review

A. Machine Learning for Fake News Detection

Researchers have actively explored diverse machine learning techniques, particularly **Natural Language Processing (NLP)**, for automated fake news detection. Studies leveraging pre-trained language models like **BERT** (Aljawarneh & Swedat, 2022) demonstrate notable accuracy improvements, highlighting the potential for enhanced contextual understanding. Additionally, **ensemble approaches** combining multiple machine learning models have shown promise in achieving robust and reliable detection (Seetharaman et al., 2022).

B. Browser Plugins: Bringing Detection to Users

The integration of machine learning algorithms into **browser plugins** offers a direct and user-centric approach to combating fake news. Solutions like **Check-It** (Paschalides et al., 2021) employ diverse signals to analyze content and

provide real-time feedback to users. By adhering to privacy regulations and offering user-friendly interfaces, these plugins aim to equip individuals with the tools to critically evaluate online information.

C. Limitations and Future Directions

Despite their potential, existing solutions face certain limitations. Prevalent biases in datasets and algorithms raise concerns about fairness and generalizability (Kula et al., 2020). While some studies lack practical implementation strategies, others require further investigation of scalability and real-time performance (Jain & Kasbe, 2018; Aljawarneh & Swedat, 2022; Zhang et al., 2023). Future research should address these limitations, focusing on building robust, scalable, and user-friendly solutions that adapt to the dynamic nature of fake news dissemination.

III. Methodology

This section outlines the methodology employed to investigate the effectiveness of various Natural Language Processing (NLP) techniques in detecting fake news. We adopted a supervised machine learning approach and explored four well-established algorithms: Support Vector Machines (SVM), Naive Bayes, Random Forest, and Logistic Regression.

A. Hardware and Software Framework

For all case scenarios, an NVIDIA Tesla K80 with 12GB of VRAM running Python 3.9.0 on a Linux machine was utilized. Pytorch and Scikit Learn are the machine learning frameworks used for building and evaluating the models.

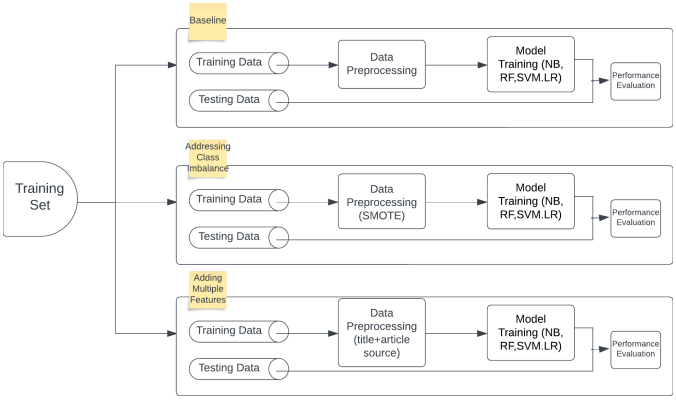

Fig. 1: Training Pipeline Architecture

B. Dataset and Preprocessing

We utilize the FakeNewNet dataset, sourced from PolitiFact and GossipCop, which provides labeled samples of real and fake news (Shu et al., 2018). After merging the datasets, we extract pertinent features such as article content and news sources. To enhance data quality, we conduct cleaning procedures by eliminating stopwords and punctuation. Subsequently, we employ CountVectorizer and TFIDF Vectorizer, as experimented in (Raza et al., 2021), to convert textual data into numerical formats conducive to machine learning analysis. Lastly, we partition the dataset into training (80%) and testing (20%) sets for model training and evaluation purposes.

C. Model Building, Evaluation and Baselines

We evaluate four classification algorithms: Naive Bayes, Random Forest, Support Vector Machine, and Logistic Regression. These algorithms are chosen based on their effectiveness in text classification tasks, as demonstrated by previous research (Fodeh et al., 2021; Bandeh and Declan, 2020). We perform hyperparameter tuning for each algorithm to optimize its performance. The model evaluation metrics include accuracy, precision, recall, and F1-score.

We provide a concise overview of Approach 1, which serves as a baseline for subsequent comparisons. It details the dataset creation process, data preparation steps, and the chosen baseline model. Detailed information about data sources, cleaning procedures, and feature engineering techniques is provided.

D. Browser Plugin Integration

The ultimate goal is to integrate the most effective model into a user-friendly browser plugin. This plugin will analyze news articles in real-time and provide users with a visual indication of potential fake news content. The key steps involved, first, developing a backend system to host the trained model, enabling it to process and classify news articles dynamically. Next, we created the Chrome browser plugin using modern web technologies such as JavaScript, HTML, and CSS. The plugin interfaces with the backend system via API calls, sending the text of news articles for classification and receiving the results instantaneously.

E. Further Exploration

We plan to explore additional NLP techniques and algorithms, such as deep learning models with pre-trained representations like BERT. We will also investigate ensemble methods to improve overall detection accuracy potentially. The results of these explorations will be presented and compared to the baseline approach.

IV. Results

Approach 1

The model's performance metrics, including accuracy, F1 score, precision, and recall, were measured for each classifier and text feature technique (Count and TFIDF). The results are summarised below:

TABLE I: Model performance using different classifiers

Model	Preprocess	Acc. (%)	F1	Prec.	Rec.	ROC
SVM	Count	79	0.86	0.87	0.86	0.90
SVM	TFIDF	86	0.91	0.87	0.96	0.94
Naive Bayes	COUNT	78	0.85	0.90	0.81	0.92
Naive Bayes	TFIDF	78	0.87	0.78	1.00	0.87
Rand. Forest	COUNT	85	0.91	0.85	0.85	0.65
Rand. Forest	TFIDF	84	0.91	0.84	0.99	0.66
Logistic Reg.	COUNT	82	0.89	0.87	0.90	0.81
Logistic Reg.	TFIDF	85	0.91	0.85	0.98	0.86

These results provide a comprehensive overview of the model's performance using different classifiers and text feature techniques in the classification task.

Approach 2: Model Enhancement with Addressed Class Imbalance

1) Creating the Dataset: Building upon the insights gained in Approach 1, Approach 2 takes a deeper dive into model improvement by addressing the class imbalance issue detected during the data preparation phase. In Approach 1, we recognized that the class imbalance could potentially bias our model's performance, emphasising the need for corrective measures.

To tackle this challenge, we implemented the Synthetic Minority Over-sampling Technique (SMOTE) as an oversampling method. SMOTE strategically targets the minority class within the dataset, generating synthetic examples to rebalance the class distribution (Chamseddine et al. 2022). This rebalancing ensures that all classes are equally represented, mitigating potential bias and enhancing the overall robustness of our model.

2) Building the model: In Approach 2, we retain the modelling approach from Approach 1, building upon the foundation laid in the initial model development. Below, we present the results obtained from this refined model, showcasing the improvements achieved in accuracy, F1 score, precision, and recall across various text extraction techniques and classifiers.

The model's performance metrics, including accuracy, F1 score, precision, and recall, were measured for each classifier and text feature technique (Count and TFIDF). The results are summarised below:

TABLE II: Classifier Performance Variation Following Data Imbalance Mitigation

Model	Preprocess	Acc. (%)	F1	Prec.	Rec.	ROC
SVM	Count	85	0.84	0.88	0.81	0.89
SVM	TFIDF	87	0.87	0.90	0.84	0.93
Naive Bayes	Count	83	0.83	0.83	0.83	0.87
Naive Bayes	TFIDF	77	0.81	0.70	0.95	0.90
Rand. Forest	TFIDF	92	0.92	0.90	0.95	0.92
Rand. Forest	Count	87	0.88	0.85	0.91	0.87
Logistic Reg.	Count	87	0.86	0.89	0.84	0.91
Logistic Reg.	TFIDF	85	0.85	0.86	0.83	0.92

Approach 3: Constructing Machine Learning Models Using Multiple Textual Features

1) Creating the Dataset: In addition to the steps taken in Approach 2 for dataset creation, we went a step further by performing additional data extraction. To be more specific, we gathered two additional text features: "article title" and "news source," extracted from the original dataset's URL column.

These text features were then combined with the content title. The goal behind extracting these text features was to boost the model's accuracy by integrating them into the modelling process.

2) Building the model: In Approach 3, we retain the modelling approach from Approach 1 , building upon the foundation laid in the initial model development. Below, we present the results obtained from this refined model, showcasing the improvements achieved in accuracy, F1 score, precision, and recall across various text extraction techniques and classifiers.

TABLE III: Evaluation of Model Performance Across Various Classifiers Using Multiple Features

Model	Preprocess	Acc. (%)	F1	Prec.	Rec.	ROC
SVM	Count	86	0.85	0.89	0.82	0.90
SVM	TFIDF	90	0.90	0.92	0.89	0.96
Naive Bayes	Count	84	0.84	0.84	0.84	0.88
Naive Bayes	TFIDF	83	0.82	0.89	0.75	0.90
Rand. Forest	Count	88	0.89	0.86	0.92	0.88
Rand. Forest	TFIDF	91	0.91	0.89	0.94	0.91
Logistic Reg.	Count	87	0.87	0.89	0.85	0.92
Logistic Reg.	TFIDF	88	0.88	0.87	0.89	0.95

3) Building the model: In this study's methodology, we utilize a Transformer-based text classifier, named Transformer-Classifier, within the PyTorch framework, to perform text classification. This custom class integrates essential components necessary for efficient text classification using the transformer architecture. We adopt the transformer classifer architecture and its components for training the model (Ashish et al., 2017)

This methodology provides a foundation for training and evaluating a Transformer-based text classifier, emphasizing key architectural components, hyperparameters, and data processing steps essential for effective text classification. Ashish et al. clearly describe this architecture.

The results are summarised below:

Hyperparameter	**Exp. 1**	**Exp. 2**	**Exp. 3**	**Exp. 4**
Epochs	50			
Learning Rate	0.0005			
Batch Size	16	32	16	32
Max Length	10000			
Number of Heads	4	4	2	2
Optimizer	Adams			
Accuracy (%)	83.6	82.9	83.5	85.8

TABLE IV: Transformer Hyperparameter Configuration

V. Evaluation & Analysis of Results

In our analysis, we examined the performance of various machine learning models through four distinct approaches, all aimed at enhancing news classification. These approaches encompassed strategies for handling class imbalance, incorporating supplementary textual features, and experimenting with transformer-based models. We will now delve into the outcomes of the top-performing model based on the results presented. We also discuss the outcome from incorporating the models as a browser plugin.

In approach 2, we tackled class imbalance using the SMOTE technique and assessed the models. Key takeaways from Table 2 include:

- Random Forest (TFIDF) emerged as the top-performing model, boasting an impressive accuracy of 92% and high F1 score, precision, and recall.
- SVM (TFIDF) continued to perform well, maintaining an accuracy of 87%.
- Logistic Regression (Count) and Logistic Regression (TFIDF) also yielded robust results.
- Naive Bayes (TFIDF) displayed a significant performance improvement compared to approach 1.

Also shown below are the confusion matrices for the models by text extraction.

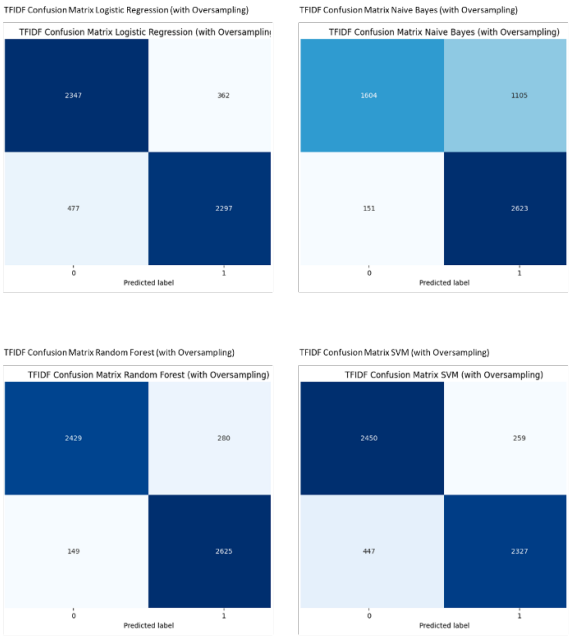

Fig. 2: Confusion Matrices Utilising TFIDF Vectorization Following Data Imbalance Mitigation

Logistic Regression + COUNTVectorizer + SMOTE

- High TN and TP counts indicate good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts indicate a balanced performance.
- SMOTE appears to effectively address the challenge of imbalanced data within the model.

Logistic Regression + TFIDFVectorizer + SMOTE

- High TN and TP counts, suggesting good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts, indicating a balanced performance.
- SMOTE helps in improving model performance, similar to the previous case.

Random Forest + CountVectorizer + SMOTE

- High TN and TP counts, indicating good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts, indicating a balanced performance.

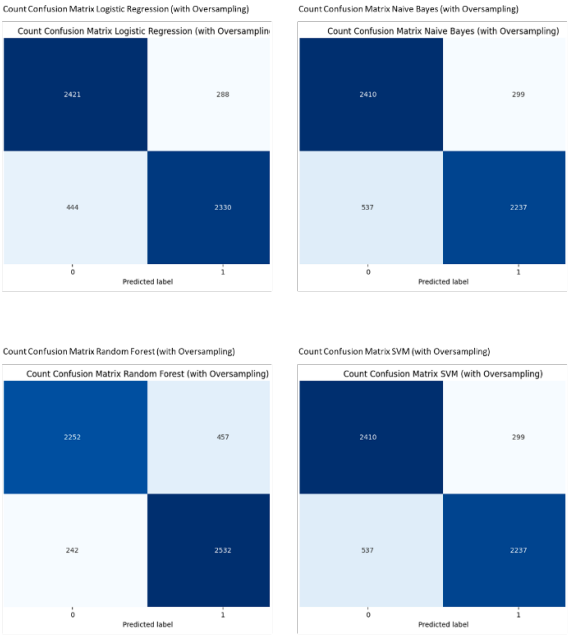

Fig. 3: Confusion Matrices Utilising Count Vectorization Following Data Imbalance Mitigation

- SMOTE helps mitigate the class imbalance issue and improves overall performance.

Random Forest + TFIDFVectorizer + SMOTE

- High TN and TP counts, indicating good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts, indicating a balanced performance.
- SMOTE has a positive impact on the model's performance.

Naive Bayes + TFIDFVectorizer + SMOTE

- Moderate TN and TP counts.
- High FP count, suggesting a tendency to misclassify negatives as positives.
- SMOTE helps improve the recall but may still result in some false positives.

Naive Bayes + CountVectorizer + SMOTE

- High TN and TP counts, indicating good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts, indicating a balanced performance.
- SMOTE is effective in improving overall model performance.

SVM + CountVectorizer + SMOTE

- High TN and TP counts indicate good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts indicate a balanced performance.
- SMOTE helps in handling class imbalances.

SVM + TFIDFVectorizer + SMOTE

- High TN and TP counts, indicating good performance in correctly classifying both negative and positive cases.
- Low FP and FN counts, indicating a balanced performance.
- SMOTE contributes to improving the model's overall performance.

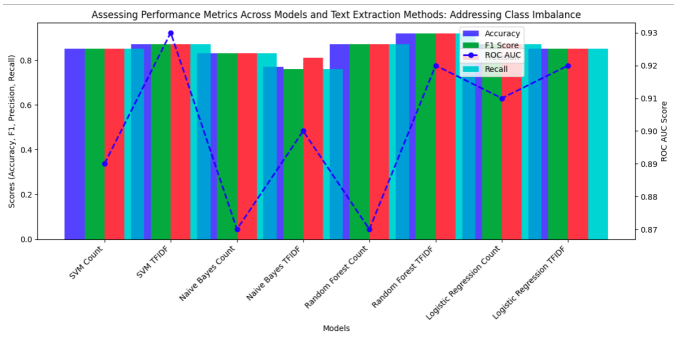

Fig. 4: Evaluation Metrics of Models Following Class Imbalance Mitigation

Comparison of Results with Existing Research

In this section, we provide a comprehensive comparison of the results obtained in our study with those reported by Shu et al. (2018), specifically focusing on fake news detection. We present the findings from each of our best-performing algorithms and align them with relevant aspects of the literature's methodology and outcomes.

		Methodological Difference		
	Model	Accuracy (%)	Imbalance Handling	Text Extraction
Our Result	Random Forest	92.0	SMOTE	TFIDF
Existing Research	Logistic Regression	82.2	-	Auto-encoders with LSTM

TABLE V: Result Comparison with Existing Research

In contrast to the existing literature, our approach involved the merging of datasets from two distinct sources. While prior studies, such as FakeNewsNet, were trained independently on PolitiFact and GossipCop datasets, our work sought to unify these sources, a step that we believe would have allowed the model to classify more effectively. Furthermore, our methodology addressed the issue of class imbalance explicitly, employing the SMOTE technique, a departure from the literature's omission in this regard. Additionally, we incorporated TF-IDF as a text extraction method, a departure from the literature's approach, which did not utilise this technique.

A. Further evaluation of model through integration with Browser plugin

The developed plugin provides an interpretation of the scraped news article content, indicating whether it is genuine or fake. Moreover, it includes a percentage probability that quantifies the extent to which the content aligns with either real or fake characteristics. If the percentage probability falls below 50%, the result is classified as fake. Feedback from user tests assessed the Installation Process, User Interface, and and accuracy of fact-checking.

Some feedback from some users

1) *I found it quite easy to install the plugin.*
2) *The plugin's user interface is intuitive and user-friendly. I didn't have to guess where to click or what to do.*
3) *The output displaying the probability of the content being real or fake was clear and easy to understand.*
4) *The design could be more visually appealing.*
5) *I am satisfied with the plugin's performance. It's a valuable tool for identifying fake news.*

VI. Conclusion

In this comprehensive analysis and evaluation of machine learning models for news classification, we explored various approaches to enhance the accuracy and robustness of our model. We began by evaluating initial models, progressing through addressing class imbalance, incorporating multiple textual features, and experimenting with transformer-based models.

Approach 1, our initial model evaluation, provided us with valuable insights into conventional classifiers' performance. SVM (TFIDF) emerged as a robust choice with an accuracy of 86%, demonstrating the potential of utilising TFIDF vectorization for feature extraction.

Approach 2, which addressed class imbalance using the SMOTE technique, yielded promising results. Random Forest (TFIDF) emerged as the top-performing model, boasting an impressive accuracy of 92%.

Approach 3 involved incorporating multiple textual features, significantly improving model accuracy. Random Forest (TFIDF) led the way with an accuracy of 91%, highlighting the advantages of additional features.

Approach 4 focused on analysing different hyperparameter configurations for transformer-based models. Experiment 4, with 50 epochs, a learning rate of 0.0005, a batch size of 32, 2 attention heads, and the "Adams" optimizer, achieved the highest accuracy at 85.8%, providing insights into hyperparameter tuning.

Upon thorough evaluation, Random Forest (TFIDF) from Approach 2 emerged as the best model candidate, with an accuracy rate of 92% and commendable performance across various metrics.

VII. Future Works

1) Multilingual Support: Enhancing the plugin's functionality to encompass multiple languages represents a vital direction for ongoing research. Given that misinformation is a worldwide concern, the capacity to identify false information in languages beyond English is of utmost importance. Future studies could explore the challenges and opportunities of implementing multilingual fake news detection, considering linguistic nuances and cultural contexts.

2) Bias Mitigation: Fake news detection models can inadvertently perpetuate biases present in training data. Further research could focus on developing methods to mitigate bias in fake news detection algorithms, ensuring fair and equitable evaluations of news content, regardless of the subject matter.
3) Multimodal Analysis: Fake news often includes images, videos, and audio content. Investigating techniques for multimodal fake news detection, where the plugin can analyse text as well as visual and auditory content, would provide a more comprehensive solution for identifying misinformation.

REFERENCES

1. Aljawarneh, S. A., & Swedat, S. A. (2022). Fake News Detection Using Enhanced BERT. IEEE Transactions on Computational Social Systems. doi:10.1109/TCSS.2022.3223786.

2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaxiser, and Illia Polosukhin. (2023). Attention Is All You Need.

3. Bandeh A. and Declan O. (2020). 'Multi-Class Imbalance in Text Classification: A Feature Engineering Approach to Detect Cyberbullying on Twitter.' Informatics, 7(4), DOI:10.3390/informatics7040052.

4. Chamseddine, E. et al. (2022). Handling class imbalance in COVID-19 chest X-ray images classification: Using SMOTE and weighted loss. Applied Soft Computing, 129, 109588. https://doi.org/10.1016/j.asoc.2022.109588.

5. Fodeh, S., et al. (2021). 'Utilizing a multi-class classification approach to detect therapeutic and recreational misuse of opioids on Twitter. Computers in Biology and Medicine, 129, 104132. ISSN0010-4825. https://doi.org/10.1016/j.compbiomed.2020.104132.

6. Jain, A., & Kasbe, A. (2018). Fake News Detection. In IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-5). Bhopal, India. doi: 10.1109/SCEECS.2018.8546944.

7. Kula, S., et al. (2020). Sentiment Analysis for Fake News Detection by Means of Neural Networks. In: Krzhizhanovskaya, V.V., et al. Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science(), vol 12140. Springer, Cham. https://doi.org/10.1007/978-3-030-50423-6_49.

8. Paschalides, D., et al. (2021). Check-It: A plugin for detecting fake news on the web. Online Social Networks and Media, 25, 100156. ISSN 2468-6964. https://doi.org/10.1016/j.osnem.2021.100156.

9. Qin, Zhang, et al. (2023). A Deep Learning-based Fast Fake News Detection Model for Cyber-Physical Social Services. Pattern Recognition Letters, 168, 31-38. ISSN 0167-8655. https://doi.org/10.1016/j.patrec.2023.02.026.

10. Raza, G. M., Butt, Z. S., Latif, S., & Wahid, A. (2021). Sentiment Analysis on COVID Tweets: An Experimental Analysis on the Impact of Count Vectorizer and TF-IDF on Sentiment Predictions using Deep Learning Models. In International Conference on Digital Futures and Transformative Technologies (ICoDT2) (pp. 1-6). Islamabad, Pakistan. doi: 10.1109/ICoDT252288.2021.9441508.

11. Seetharaman, R., Tharun, M., Sreeja Mole, S. S., & Anandan, K. (2022). Analysis of fake news detection using machine learning technique. Materials Today: Proceedings, 51(Part 8), 2218-2223. ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2021.11.334.

12. Shu, K., et al. (2018). FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. arXiv preprint arXiv:1809.01286.

13. Senthil Raja, M., & Arun Raj, L. (2022). Fake news detection on social networks using Machine learning techniques. Materials Today: Proceedings, 62(7), 4821-4827. ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2022.03.351.

14. TR, R., Lilhore, U. K., M, P., Simaiya, S., Kaur, A., & Hamdi, M. (2022). Predictive Analysis of Heart Diseases with Machine Learning Approaches. Malaysian Journal of Computer Science, 132–148. https://doi.org/10.22452/mjcs.sp2022no1.10.