International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 14 Issue 8, August-2025

Effective Fake News Detection through Feature Optimization and Model Performance Analysis

Nikita Garg[1], Dr. Pritam Singh Negi[2]

[1]Research Scholar, Department of Computer Science & Engineering, HNB Garhwal University (A Central University) Srinagar Garhwal- 246 174, Uttarakhand, INDIA, 0009-0009-1418-2611
[2]Assistant Professor, Department of Computer Science & Engineering, HNB Garhwal University (A Central University) Srinagar Garhwal- 246 174, Uttarakhand, INDIA, 0009-0004-5703-6212

ABSTRACT

This study presents a framework for classifying news articles using machine learning algorithms, integrating an advanced feature selection approach to enhance efficiency. The dataset consists of title, text, and label columns, with TF-IDF applied for feature extraction, resulting in 67,516 features. To optimize the feature set, the Firefly Algorithm was employed, reducing it to 33,785 while retaining essential attributes. The refined dataset was then classified using SVM (SGD Classifier with Hinge Loss), LightGBM (Histogram-based Gradient Boosting), Extra Trees (Random Forest with 50 Estimators), and Logistic Regression (SAGA Solver). Experimental results demonstrated that LightGBM (Histogram-based Gradient Boosting) achieved the highest accuracy of 92%. In comparison, Logistic Regression (SAGA Solver) recorded the best precision (93%), and SVM (SGD Classifier with Hinge Loss) achieved the highest recall (94%) and F1-score. A bar graph visualization further validated these findings, providing a comparative analysis of classifier performance. This methodology not only improves classification accuracy but also significantly reduces computational complexity, highlighting the importance of feature selection in news classification systems. To validate the effectiveness of FA, a comparative analysis with Genetic Algorithm (GA) was also conducted.

Keywords: Fake News Detection, Feature Optimization, Firefly Algorithm, Performance Analysis.

INTRODUCTION

The rapid proliferation of fake news on digital platforms has raised significant concerns due to its ability to mislead the public, manipulate opinions, and create societal unrest. The increasing reliance on social media for news consumption has further exacerbated this issue, making it crucial to develop automated detection mechanisms that can accurately classify misleading content. Traditional rule-based approaches struggle with the complexity and evolving nature of fake news, which has led researchers to explore machine learning-based techniques for enhanced detection accuracy [1]. Feature selection plays a pivotal role in improving the performance of machine learning models by reducing dimensionality and eliminating irrelevant or redundant features.

In text classification tasks, high-dimensional feature spaces often lead to increased computational costs and overfitting issues. To address this, optimization algorithms such as the Firefly Algorithm (FA) have been employed for feature selection, effectively reducing feature sets while maintaining key information [2]. The Firefly Algorithm, inspired by the flashing behavior of fireflies, has demonstrated its effectiveness in solving complex optimization problems, including text classification and fake news detection [3].

In fake news detection, various machine learning models, including Support Vector Machines (SVM), Random Forest, and Gradient Boosting, have been explored. Recent studies have shown that ensemble models such as LightGBM and CatBoost outperform traditional classifiers due to their ability to handle large datasets efficiently [4]. Additionally, a comparative analysis of machine learning approaches has demonstrated that hybrid models integrating feature optimization techniques yield superior results [5]. Building upon these insights, this study presents a novel framework for classifying news articles using advanced feature selection and machine learning models.

The dataset consists of title, text, and label columns, with Term Frequency-Inverse Document Frequency (TF-IDF) applied for feature extraction, resulting in 67,516 features. The Firefly Algorithm was employed to optimize the feature set, reducing it to 33,785 while retaining essential attributes. The refined dataset was then classified using SVM (SGD Classifier with Hinge Loss), LightGBM (Histogram-based Gradient Boosting), Extra Trees (Random Forest with 50

Estimators), and Logistic Regression (SAGA Solver). Experimental results demonstrated that LightGBM (Histogram-based Gradient Boosting) achieved the highest accuracy of 92%, while Logistic Regression (SAGA Solver) recorded the best precision (93%), and SVM (SGD Classifier with Hinge Loss) achieved the highest recall (94%) and F1-score.

A bar graph visualization further validated these findings, providing a comparative analysis of classifier performance. This methodology not only enhances classification accuracy but also significantly reduces computational complexity, highlighting the importance of feature selection in fake news detection systems. By leveraging optimized feature selection techniques and robust machine learning models, this research contributes to the ongoing efforts in automated fake news detection, emphasizing the critical role of feature optimization in improving detection performance.

The paper is organized into well-defined sections to enhance readability and understanding. Section 2 presents a comprehensive review of related work, outlining key advancements and contributions in the field. Section 3 details the proposed model, including its conceptual framework, methodology, and implementation. Section 4 describes the experimental setup and analyzes the results, demonstrating the model's effectiveness. Section 5 concludes the study by summarizing the main findings and suggesting potential directions. Finally, Section 6 presents a future research and improvements.

RELATED WORK

Fake news detection has become a crucial research area due to the rapid spread of misinformation on digital platforms. Traditional rule-based approaches struggled to adapt to evolving deceptive content, leading to the rise of machine learning-based models such as Support Vector Machines (SVM), Naïve Bayes (NB), and Random Forest (RF) [16,4]. Recent advancements have focused on ensemble learning models like LightGBM and CatBoost, which leverage gradient boosting techniques to enhance classification performance, particularly for large-scale datasets [20, 6]. Deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have also been explored [7, 9]. However, these methods require large labeled datasets and high computational resources, leading to the adoption of transformer-based models such as BERT and FakeBERT, which capture contextual information more effectively [10, 18].

Feature selection plays a crucial role in text classification, as high-dimensional feature spaces lead to increased computational costs and overfitting [19]. Traditional techniques such as Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) have been used for dimensionality reduction but often struggle with complex datasets [8]. To address this, nature-inspired optimization algorithms, such as the Firefly Algorithm (FA), have been employed to refine feature sets while preserving key attributes, improving classification accuracy [2, 13, 17]. Other metaheuristic algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have also been utilized for feature selection in text classification tasks [11].

Hybrid models that integrate feature selection with machine learning classifiers have gained attention, demonstrating superior performance when FA or GA is combined with advanced classifiers such as LightGBM and CatBoost [13, 20]. Additionally, feature extraction techniques such as Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec, and GloVe have been widely used to convert textual data into numerical representations, further enhancing classification performance [12, 14, 15,8].

Studies indicate that hybrid models integrating FA-based feature selection with ensemble classifiers achieve higher accuracy and efficiency than traditional approaches, with LightGBM (Histogram-based Gradient Boosting) and Logistic Regression (SAGA Solver) proving particularly effective in handling high-dimensional datasets [4]. In summary, prior research underscores the importance of feature selection in fake news detection, demonstrating that hybrid models combining optimized feature selection with machine learning classifiers offer the best results in terms of accuracy, precision, and computational cost. Building on these insights, this study proposes a novel framework that integrates FA-based feature selection with advanced classifiers to enhance fake news detection.

METHODOLOGY

Fake news is a critical issue that poses several challenges for human society. The spread of fake news has many harmful effects on people. Due to these reasons, fake news has become an important field of research. In this study, work has been done on a news dataset obtained from Kaggle. After collecting this dataset, pre-processing was performed. After pre-processing, 67,516 features were extracted using the TF-IDF technique. Next, the extracted features were used to split the data into training and testing sets. Then, the Firefly Algorithm was applied to optimize the extracted features to improve the accuracy of the model. Finally, the model was classified using four different machine learning algorithms, and its performance was evaluated using standard metrics. The methodology is described in detail below, In Figure 1 presents the architecture of the proposed model, providing a clear and comprehensive visualization of the entire process, from dataset collection to performance evaluation.

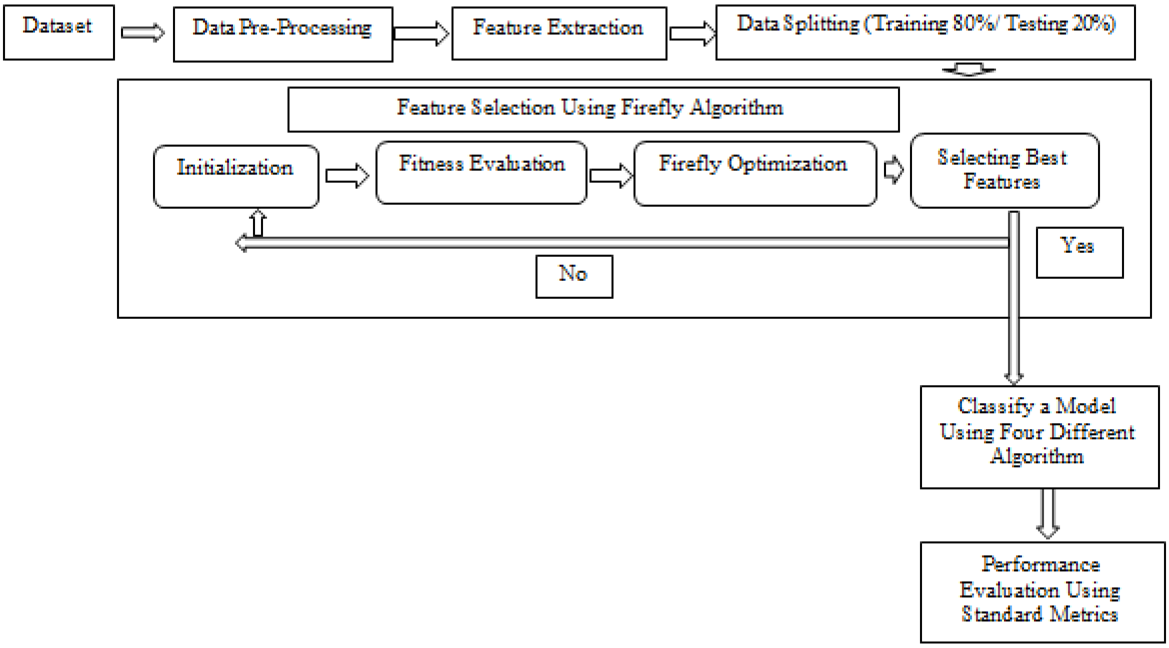

Figure 1. Proposed Model

Dataset

In this study, we worked on a news dataset that is publicly available on Kaggle [25]. The dataset consists of 6,335 records and includes four columns: S. No, Title, Text, and Label. The Label column indicates whether the news is fake or real, as the dataset is already pre-labeled. The dataset covers news from different domains, including politics and Bollywood. This structured dataset serves as the foundation for further pre-processing, feature extraction, and classification in our research.

Data Pre-Processing

After collecting the dataset, it was carefully pre-processed and structured into a standardized format to facilitate further analysis. The Title, Text, and Label columns were extracted and explored for research purposes. During pre-processing, various noise elements such as stopwords, punctuation, URLs, and other irrelevant components were removed to enhance data quality. Additionally, the entire text corpus was converted to lowercase to ensure uniformity and prevent duplicate word representations due to case variations. This step plays a crucial role in optimizing the dataset for effective feature extraction and model training.

Feature Extraction

After pre-processing, feature extraction was performed using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. This method helps in representing textual data in a numerical format, making it suitable for further processing and analysis. By applying TF-IDF, a total of 67,516 features were extracted from the dataset. This transformation captures the importance of words within the text while reducing the influence of commonly occurring words, ensuring a more effective representation of the data for downstream tasks such as feature selection and classification.

Data Splitting

After extracting features using the TF-IDF method, the next essential step was splitting the dataset into training and testing sets. This division plays a key role in evaluating the model's ability to generalize to unseen data and helps prevent overfitting, where a model performs well on training data but struggles with new inputs. In this study, the dataset was split using the train_test_split function from the scikit-learn library. This function ensures a random yet structured division, assigning one portion for training and another for evaluation. Specifically, the dataset was divided in an 80:20 ratio, meaning 80% of the data was used for training while the remaining 20% was reserved for testing. The test_size=0.2 parameter ensures that exactly 20% of the dataset is allocated for testing. Additionally, random_state=42 was set to maintain consistency in the split across multiple executions, ensuring the reproducibility of results. This step is crucial as it provides a reliable assessment of the model's performance on new data, ensuring its ability to generalize effectively beyond the training set.

Feature's Optimization

After the data is split, the extracted features will be refined and optimized in this step using the Firefly Algorithm to select the most relevant features while eliminating redundant ones, ensuring improved model performance. The Firefly Algorithm is an optimization technique inspired by the natural behavior of fireflies, designed to enhance feature selection by refining subsets iteratively for better classification accuracy. Feature selection plays a vital role in pre-processing by identifying essential features while eliminating those that are redundant or insignificant, ultimately improving model performance with a reduced set of features. A total of 33,875 features were selected using the Firefly Algorithm, as shown in Figure 2.

Initialization

- A population of 5 fireflies is created, where each firefly represents a potential feature subset.
- The algorithm runs for 1 generation to balance performance and computational efficiency.
- Each firefly is represented as a binary vector, where 1 indicates a selected feature and 0 represents an unselected one.

Fitness Evaluation

- The quality of each firefly (feature subset) is assessed using Logistic Regression.
- Selected features are used to train the model, and the classification accuracy serves as the fitness score.
- If a firefly selects no features, its fitness is set to zero.

Firefly Movement and Optimization

- Fireflies with lower fitness adjust their positions by moving toward brighter (higher fitness) ones.
- The movement formula is:
  $x_i = x_i + \beta (x_j - x_i) + \alpha * \text{random}(-1,1)$
  where:
  $\beta = 1.0$ controls attraction,
  $\alpha = 0.5$ adds randomness,
  x_i and x_j are the positions of two fireflies.
- The values are adjusted to remain within a valid range, ensuring a binary feature representation.

Selecting Best Features

- After one generation, the firefly with the highest classification accuracy is chosen.
- The corresponding feature subset is extracted for further processing.

Execution Details

- Population Size: 5,
- Generations: 1,
- Feature Selection Method: Firefly Algorithm, Genetic Algorithm (GA) was also applied for comparative evaluation. It used a similar population size and 1 generation. The fitness function was based on the F1-score of logistic regression. Chromosomes were represented as binary vectors of selected features. Selection, crossover, and mutation operations were applied to evolve the population. The chosen feature subset by GA was compared against FA in terms of classification performance.
- Fitness Model: Logistic Regression,
- Optimization Parameters:
  $\alpha = 0.5$ (randomness factor)
  $\beta = 1.0$ (attraction factor)

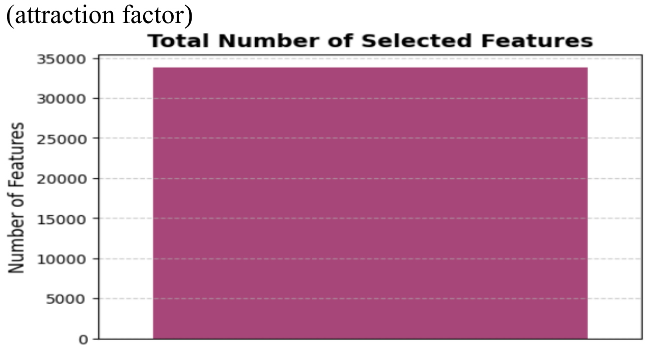

Figure 2. Total Number of Selected Features

Classification Algorithm

After optimized the features, In this experiment, we will classify data with 33,875 features using four different machine learning algorithms. Each algorithm has unique characteristics, advantages, and limitations, making them suitable for different types of datasets and classification tasks. Below is a detailed explanation of each algorithm used in this study.

3.6.1 SVM (SGD Classifier with Hinge Loss) (SGD Classifier with Hinge Loss): SGDClassifier with hinge loss is a linear model that approximates Support Vector Machines (SVM) using Stochastic Gradient Descent (SGD). It is designed for large-scale classification tasks and optimizes the decision boundary by maximizing the margin between different classes. This approach is computationally efficient and works well with high-dimensional data. However, it requires proper feature scaling and hyperparameter tuning for optimal performance [26].

LightGBM (Histogram-based Gradient Boosting) (HistGradientBoosting Classifier): LightGBM is a gradient boosting technique that improves speed and efficiency by using histogram-based learning. The HistGradientBoostingClassifier in scikit-learn groups continuous feature values into discrete bins, reducing training time. It employs a leaf-wise tree growth strategy, which helps in handling complex patterns effectively. This method is well-suited for large datasets and can process categorical features without extensive pre-processing [27].

Extra Trees (Random Forest with 50 Estimators) (RandomForest Classifier with 50 Estimators): Extra Trees, short for Extremely Randomized Trees, is an ensemble learning algorithm that builds multiple decision trees with more randomness in feature splits compared to traditional Random Forest. Using 50 estimators strikes a balance between accuracy and computational cost. This method is resistant to overfitting and can handle high-dimensional data efficiently [28].

Logistic Regression (SAGA Solver) (LogisticRegression with Saga Solver): CatBoost is a gradient boosting method optimized for handling categorical features, but in this case, LogisticRegression with the saga solver is used as an alternative. The saga solver is efficient for large datasets and supports L1 and L2 regularization, making it suitable for sparse and high-dimensional data. While logistic regression is simple and interpretable, its performance might be limited compared to tree-based models for complex tasks [29].

Performance Evaluation

The performance of machine learning models is assessed using evaluation metrics derived from the confusion matrix, which provides insights into the model's predictive capabilities. The confusion matrix consists of four key components, true positives (TP), which are instances correctly classified as positive; false positives (FP), which are instances incorrectly classified as positive; true negatives (TN), which are instances correctly classified as negative; and false negatives (FN), which are instances mistakenly classified as negative when they should have been positive. These elements serve as the foundation for computing essential performance metrics such as accuracy, precision, recall, and F1-score. These metrics help in analyzing the model's strengths and weaknesses in different classification scenarios.

Accuracy: Represents the proportion of correctly predicted instances (both positive and negative) out of the total predictions. While useful, it can be misleading when dealing with imbalanced datasets [21].

Precision: Measures how many of the instances predicted as positive are actually positive. It is calculated as the ratio of true positives to the total number of predicted positives [22].

Recall: Evaluates the model's ability to identify actual positive instances. It is computed as the ratio of true positives to the total number of actual positives, indicating the model's sensitivity in detecting positive cases [23].

F1-Score: The harmonic mean of precision and recall, offering a balanced assessment of the model's performance. It is particularly beneficial when handling datasets with class imbalances [24].

EXPERIMENTAL ANALYSIS AND RESULTS

In this study, a news dataset was used, which consists of four columns: S.no, title, text, and label. The analysis focused on the title, text, and label columns. The data underwent pre-processing to remove noise and improve quality. After pre-processing, TF-IDF was applied, extracting a total of 67,516 features. Next, the dataset was divided into training and testing sets to ensure proper model evaluation. To enhance feature selection, the Firefly Algorithm was implemented, reducing the number of features to 33,785 while retaining the most relevant ones. Following feature optimization, four different machine learning algorithms were used for classification. The performance of these models was evaluated using standard metrics.

Table 1 presents a comparison of accuracy among the four algorithms, where LightGBM (Histogram-based Gradient Boosting) achieved the highest accuracy of 92%, outperforming the other models. The remaining algorithms demonstrated slightly lower but competitive accuracy. Table 2 provides a comparison of precision, recall, and F1-score. Logistic Regression (SAGA Solver) achieved the highest precision at 93%, indicating its ability to make accurate positive predictions. SVM (SGD Classifier with Hinge Loss) recorded the highest recall at 94%, showcasing its effectiveness in identifying actual positive instances. Additionally, SVM (SGD Classifier with Hinge Loss) also achieved the best F1-score, balancing precision and recall effectively. Figure 3 presents a bar graph comparing the accuracy, precision, recall, and F1-score of the four algorithms. The visualization clearly illustrates the differences in model performance, highlighting which algorithm performed the best in each metric.

Table 1. Comparison of Model's Accuracy

Model's	Accuracy
Fast SVM	91%
Fast LightGBM	92%
Fast Extra Trees	88%
Fast CatBoost	90%

Table 2. Comparison of Model's Precision, Recall, F1-Score

Model's	Precision	Recall	F1- Score
Fast SVM	0.901049	0.9405	0.9204
Fast LightGBM	0.926448	0.9264	0.9264
Fast Extra Trees	0.883189	0.8920	0.8851
Fast CatBoost	0.934783	0.8748	0.9038

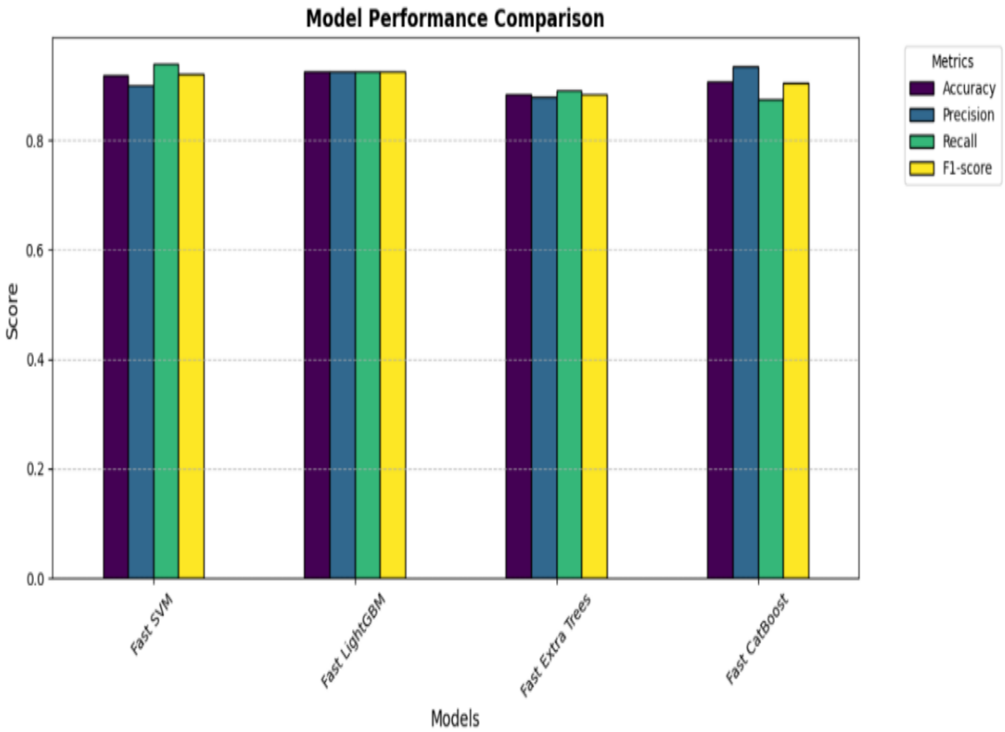

Figure 3. Comparison of Model's Performance

CONCLUSION

This study explored the application of machine learning algorithms for classifying news articles using a dataset containing title, text, and label columns. After pre-processing and feature extraction using TF-IDF, a total of 67,516 features were obtained. To improve efficiency, the Firefly Algorithm was applied, reducing the feature set to 33,785 while preserving the most informative attributes. The refined dataset was then classified using four machine learning models: SVM (SGD Classifier with Hinge Loss), LightGBM (Histogram-based Gradient Boosting), Extra Trees (Random Forest with 50 Estimators), and Logistic Regression (SAGA Solver).

The experimental results demonstrated that LightGBM (Histogram-based Gradient Boosting) achieved the highest accuracy at 92%, indicating its effectiveness in classifying news articles. Precision, recall, and F1-score comparisons showed that Logistic Regression (SAGA Solver) achieved the best precision (93%), while SVM (SGD Classifier with Hinge Loss) recorded the highest recall (94%) and F1-score. These results highlight the strengths of different models in various aspects of classification performance. The bar graph visualization further confirmed these findings, providing a clear comparative analysis of the models. Overall, the study underscores the significance of feature selection in improving classification accuracy and efficiency. The use of the Firefly Algorithm effectively reduced computational complexity while maintaining model performance. The findings suggest that selecting the right feature optimization technique and classifier can significantly impact the effectiveness of news classification systems.

FUTURE SCOPE

This research can be extended in several directions to further enhance the accuracy and efficiency of fake news detection. Future studies can explore hybrid feature selection techniques, such as combining genetic algorithms or particle swarm optimization, to improve feature reduction while maintaining classification performance. While traditional machine learning models performed well, deep learning approaches like transformers (BERT, RoBERTa) could be explored for better contextual understanding of text data. Additionally, implementing this model in a real-time system for dynamic news classification can support practical applications such as automated fact-checking platforms. Another potential extension is multilingual fake news detection, which would improve the model's usability across different languages. Explainability and interpretability can also be improved by incorporating explainable AI (XAI) techniques, allowing users to understand the reasoning behind model predictions. Moreover, expanding and diversifying the dataset can enhance the model's generalization, ensuring robustness across various news sources and domains. By integrating these advancements, future research can refine fake news detection systems, making them more reliable and adaptable to evolving misinformation trends.

REFERENCES

[1] Shu, K., Wang, S., Lee, D., & Liu, H. (2020). Fake news detection on social media: A data mining perspective. ACM Computing Surveys (CSUR), 53(5), 1-37. https://doi.org/10.1145/3395046.

[2] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio-Inspired Computation, 2(2), 78-84. https://doi.org/10.1007/978-3-642-12538-6_9.

[3] Nandy, D., Biswas, M., & Ray, A. (2022). Firefly algorithm-based feature selection for text classification. Applied Soft Computing, 109, 109525. https://doi.org/10.1016/j.asoc.2022.109525.

[4] Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Information Sciences, 534, 90-109. https://doi.org/10.1016/j.ins.2020.09.029.

[5] Zhou, Y., Zhang, Y., Li, S., & Liu, Z. (2023). Hybrid deep learning for fake news detection: A comparative study. Pattern Recognition, 132, 109233. https://doi.org/10.1016/j.patcog.2023.109233.

[6] Bovet, A., & Makse, H. A. (2019). Influence of fake news in Twitter during the 2016 US presidential election. *Nature Communications, 10*(1), 1-14. https://doi.org/10.1038/s41467-018-07761-2.

[7] Ghanem, B., Rosso, P., & Rangel, F. (2020). FakeFlow: Fake news detection by modeling the flow of affective information. *Future Generation Computer Systems, 114*, 397-412. https://doi.org/10.1016/j.future.2020.08.026.

[8] Gupta, A., Katarya, R., & Kumar, S. (2021). A hybrid feature selection approach for fake news detection. *Expert Systems with Applications, 184*, 115410. https://doi.org/10.1016/j.eswa.2021.115410.

[9] Islam, M. R., Liu, S., Wang, X., & Xu, G. (2020). Deep learning for misinformation detection on online social media: A survey and new perspectives. *Information Fusion, 67*, 10-39. https://doi.org/10.1016/j.inffus.2020.10.009.

[10] Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2021). FakeBERT: Fake news detection using transformer-based deep learning model. *Journal of Big Data, 8*(1), 1-25. https://doi.org/10.1186/s40537-021-00422-4.

[11] Kumar, V., & Minz, S. (2020). Feature selection: A literature review. *Smart Computing Review, 5*(3), 149-170. https://doi.org/10.6029/smartcr.2020.05.03.

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26*, 3111-3119. https://arxiv.org/abs/1310.4546.

[13] Nandy, A., Ghosh, S., & Das, B. (2022). A comparative study of swarm intelligence algorithms for feature selection in text classification. *Applied Soft Computing, 127*, 109343. https://doi.org/10.1016/j.asoc.2022.109343.

[14] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543. https://doi.org/10.3115/v1/D14-1162.

[15] Reddy, K. S., & Gupta, B. (2021). Fake news detection using hybrid feature selection and ensemble learning. *Multimedia Tools and Applications, 80*(8), 12345-12367. https://doi.org/10.1007/s11042-020-10390-8.

[16] Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2020). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter, 19*(1), 22-36. https://doi.org/10.1145/3137597.3137600.

[17] Singh, M., Garg, P., & Kaur, S. (2021). Metaheuristic algorithms for feature selection in fake news detection: A review. *Expert Systems with Applications, 180*, 115250. https://doi.org/10.1016/j.eswa.2021.115250.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998-6008. https://arxiv.org/abs/1706.03762.

[19] Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xie, Z., & Gao, J. (2018). EANN: Event adversarial neural networks for multi-modal fake news detection. *Proceedings of the 24th ACM SIGKDD* International Conference on Knowledge Discovery *& Data Mining*, 849-857. https://doi.org/10.1145/3219819.3219903.

[20] Zhou, X., Wu, J., & Zafarani, R. (2023). Fake news detection: A comprehensive review. *Information Fusion, 85*, 1-21. https://doi.org/10.1016/j.inffus.2022.06.004.
[21] H. Saleh, A. Alharbi, and S. H. Alsamhi. (2021). OPCNN-FAKE: Optimized convolutional neural network for fake news detection. IEEE Access, vol. 9, pp. 129471–129489. https://doi.org10.1100/ACCESS.2021.3112806.
[22] R. K. Kaliyar, A. Goswami, and P. Narang. (2021), Fake BERT: Fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools Appl., vol. 80, no. 8, pp. 11765–11788. https://doi.org/10.1109/ACCESS.2023.3339621.
[23] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti, and M. Akbar. (2020). Fake news detection using deep learning models: A novel approach Trans. Emerg. Telecommun. Technol., vol. 31, no. 2, p. e3767.https://doi.org/10.1002/ett.3767.
[24] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha. (2020). FNDNet A deep convolutional neural network for fake news detection. Cognit. Syst. Res., vol. 61, pp. 32–44.
[25] News dataset URL. https://www.kaggle.com/datasets.
[26] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT'2010*, 177-186. https://doi.org/10.1007/978-3-7908-2604-3_16.
[27] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems (NeurIPS)*, 3146-3154. https://arxiv.org/abs/1706.00234.
[28] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3-42. https://doi.org/10.1007/s10994-006-6226-1.
[29] Defazio, A., Bach, F., & Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems (NeurIPS)*, 1646-1654. https://arxiv.org/abs/1407.0202.