Fake News Detection with Dynamic Model Updates Based on Classifier Comparison

Maitreyee Jadhav[1], Mansi Patil[2], Preshika Giri[3], Vinay More[4] and Yogita Hande[5]
[1]*Student, Dept. of Engineering, Dr Vishwanath Karad MIT World Peace University*
[2]*Student, Dept. of Engineering, Dr Vishwanath Karad MIT World Peace University*
[3]*Student, Dept. of Engineering, Dr Vishwanath Karad MIT World Peace University*
[4]*Student, Dept. of Engineering, Dr Vishwanath Karad MIT World Peace University*
[5]*Assistant Professor, Dept. of Engineering, Dr Vishwanath Karad MIT World Peace University*

***Abstract*—In the digital age, the rapid dissemination of information through online platforms has led to a significant rise in the spread of fake news, posing serious threats to societal trust, political stability and public safety. This project focuses on the development of a machine learning-based system for automatic fake news detection. The system analyzes textual news content to classify it as real or fake using various classification algorithms. Initially, data preprocessing steps such as tokenization, stopword removal, and stemming were applied to clean the dataset. Features were extracted using techniques like TF-IDF and Count Vectorizer. Multiple models including Logistic Regression, Support Vector Machine (SVM), and Naive Bayes, were trained and evaluated for performance using metrics such as accuracy, precision, recall, and F1-score. The model with the highest accuracy is selected for deployment. To ensure the system remains effective over time, a dynamic model updating strategy is implemented, wherein the model is periodically retrained with newly labeled data. This approach not only enhances prediction accuracy but also adapts to evolving patterns in misinformation. The project demonstrates an adaptive solution for combating fake news using data-driven techniques.**

***Index Terms*—Fake News Detection, Logistic Regression, Support Vector Machine, Naïve Bayes**

I. INTRODUCTION

The proliferation of digital platforms has led to an unprecedented rise in the dissemination of fake news, resulting in widespread misinformation and public mistrust. Detecting fake news has become critical in maintaining the integrity of information consumed by the public. Existing fake news detection systems primarily focus on static model training, where models are trained on a fixed dataset and remain unchanged over time. However, the dynamic and evolving nature of misinformation demands systems that can adapt continuously.

Currently, fake news detection models suffer from performance degradation as new styles, formats, and types of misinformation emerge. Static models become outdated, leading to reduced accuracy and reliability. Furthermore, there is a noticeable gap in the industry and research: the absence of dynamic model updating mechanisms. To date, minimal efforts have been directed towards dynamically updating fake news detection models to maintain relevance with real-time data.

Addressing this gap, the present study proposes a system for dynamic model updating, wherein the model is periodically retrained with newly collected real and fake news articles. By doing so, the model remains aligned with current trends, improving predictive accuracy and robustness over time. The proposed methodology systematically integrates data collection, preprocessing, model training, evaluation, and dynamic updating to address this pressing research gap.

II. LITERATURE REVIEW

In [1] authors used GloVe embeddings with a CNN model (FNDNet) for fake news detection, a CNN-LSTM model for stance detection, and applies feature extraction with PCA and Chi-Square for dimensionality reduction.

The authors [2] combined TFIDF (global features), CNN (spatial features), and BiLSTM (temporal features) using early fusion, and classifies the result with a Fast Learning Network (FLN) for fake news detection. This The model is tested only on English Language dataset.

In this author [3] used a hybrid N-gram and LSTM model for fake news detection, deployed on a big data platform for parallel and distributed processing to achieve real-time, accurate classification. The model does not focus on language nuances such as sarcasm, satire, and ambiguous phrasing, leading to misclassification.

The author [4] fine-tunes the BERT model on a fake and real news dataset to enhance its detection capabilities, achieving 99.96% accuracy and outperforming other models. It does not address the high computational cost of BERT,making real-time large-scale detection difficult.

The authors [5] provided a survey on fake news in virtual communities and the metaverse, analyzing its manifestation in singlemodal and multimodal forms, and reviewing detection methods. It also discusses future directions for intelligent detection and information security in these environments.

The current research lacks the feature of dynamic model updating, so we are proposing a system for dynamic model updating, that is updating the model on regular intervals to keep it up to date with current news articles to produce better prediction outcomes.

III. METHODOLOGY

The paper employs a systematic methodology that includes collection of data, preprocessing, training models, evaluation, and taking the best model into consideration for dynamic model updating. We are addressing the noted research gaps of dynamic model updating in the methodology. The proposed system architecture is shown in Fig. 1.

A. Data Collection

We are collecting real news data from legitimate news websites through web scraping and labelling them as 0. The fake news is scraped through some legitimate websites that have a category for identified fake news, the rest of the fake news are created by using any GenAI model, prompting it to create a paraphrased fake version of some real news articles. The fake news articles data is labelled as 1. The details that we are scraping from news websites are: article title, description and the date on which the article was scraped. We are assuming that the news taken from legitimate news websites is real or true.

Table I. Dataset Features

Sr no.	Feature name	Description
1	title	Title if the news article
2	text	Description if the news article
3	label	Label of the news article; fake: 1 and real: 0
4	date	Date on which the article was web scrapped

B. Text Preprocessing

To prepare textual data for analysis, a preprocessing function is developed, which involves removing stop-words, tokenizing text, converting to lowercase, eliminating non alphabetic characters, and applying stemming or lemmatization. The dataset that we have created goes through a pre-processing function.

C. Feature extraction

Feature extraction from textual data can be accomplished by using word embeddings or TF-IDF vectorization (Term Frequency-Inverse Document Frequency). By converting text into numerical attributes, we can determine the semantic content and word importance of each document. In our implementation, we utilized the TfidfVectorizer from the scikit-learn library, configuring it with specific parameters to enhance feature extraction:

N-gram Range: Set to (1, 2) to capture both unigrams and bigrams, allowing the model to consider individual words and pairs of consecutive words, thereby incorporating some contextual information.

Maximum Document Frequency (max_df): Set to 0.9 to exclude terms that appear in more than 90% of the documents, as such terms are likely to be non-informative and may not contribute to distinguishing between documents.

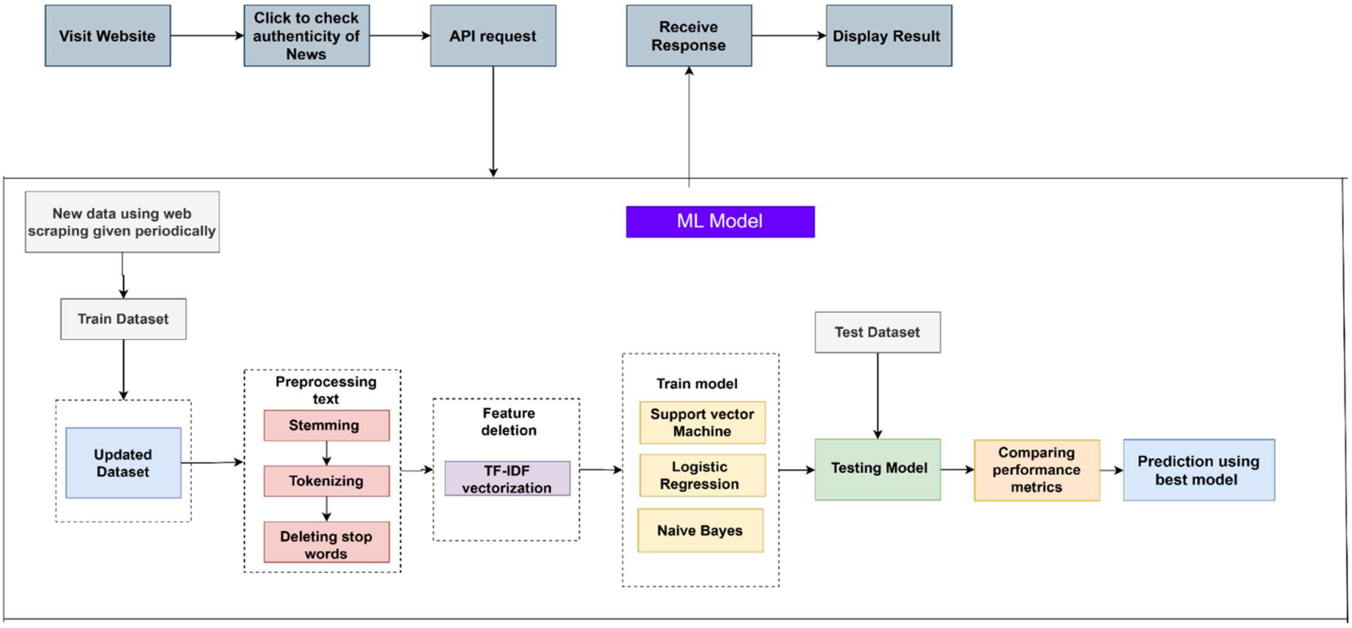

Fig. 1 Architecture Diagram

Minimum Document Frequency (min_df): Set to 2 to ignore terms that appear in fewer than two documents, reducing the impact of rare terms that may not be relevant for the analysis.

The vectorizer was fitted to the corpus and the textual data was transformed into a TF-IDF weighted term-document matrix, which served as the input for subsequent analytical processes.

D. Model Training

The pre-processed features are divided into training and testing groups using a suitable splitting technique, such as a random split or cross-validation. We have utilized random split. When selecting a machine learning model, the goals of the research and the characteristics of the dataset are taken into account. We have trained the data on three ML models, Support Vector Machine (SVM), Logistic Regression (LR) and Naïve Bayes (NB).

Support Vector Machine (SVM) is an effective classification method that searches for the best hyperplane in a multidimensional space to separate different classes. It is robust against outliers and efficiently handles both linear and non-linear decision boundaries through the use of kernel functions. By maximizing the margin between classes, SVM improves generalization on unseen data. Its adaptability to different data types and the ability to control overfitting through regularization parameters make SVM a popular choice for various classification tasks.

Logistic Regression is a widely used statistical method for binary and multiclass classification problems. It models the probability of a data instance belonging to a particular class using a logistic (sigmoid) function. The model estimates the relationship between input features and the log-odds of the outcome, making it both interpretable and computationally efficient. Logistic Regression performs best when the classes are linearly separable and includes regularization techniques to prevent overfitting, allowing it to generalize well to new data.

Naive Bayes is a simple yet powerful probabilistic classification algorithm based on Bayes' Theorem. It assumes that the features are conditionally independent given the class label, which simplifies the computation significantly. Despite its 'naive' assumption, Naive Bayes often performs competitively, particularly in text classification and spam detection tasks. It is highly efficient, requires a small amount of training data, and works well even when the independence assumption is somewhat violated in practice.

After training these models, we calculate the evaluation metrics such as accuracy, precision, recall and f1- score. The model with best accuracy is selected as the final model for predicting fake news. For dynamic model updating, we are scrapping the real news from websites on a daily basis and appending it to our dataset, and training the model weekly on this newly obtained dataset to improve the accuracy of the model based on the latest news.

IV. RESULT AND ANALYSIS

The study examines the LR, SVM and NB methodologies' performance using our dataset. The models' performance is evaluated using key performance indicators like F1-score, recall, precision, and accuracy. The method with best accuracy is used for dynamic model updating. It is observed that SVM performs better compared to LR and NB so it is selected as the final model. This SVM model is then trained periodically to increase accuracy for predicting the latest fake news. Table II, III, IV and V show the evaluation metrics for SVM such as accuracy, precision, recall and f1- score, for two weeks of data. The model was trained on new collected data once a week. The graphs show the same visualization of evaluation metrics.

The 80-20 train-test split consistently delivered the best performance overall. In Week 1, the model achieved high scores in all metrics, particularly a Recall of 0.9286 and an F1-score of 0.8387, indicating its effectiveness in identifying true positive cases and maintaining a good balance between precision and recall. Week 3 also demonstrated solid and balanced performance across all metrics, suggesting that the model performs reliably when trained on 80% of the data. However, Week 2 under this split showed a noticeable drop in recall, which may point to irregularities or noise in the data collected during that period. Table number II shows the values of accuracy, precision, recall and f1 value for week 1, week 2 and week 3 of training model with train to test data ratio is 80:20.

Table II. Performance Metrics for 80-20 Train-Test Split

Train-Test Split 80-20	Accuracy	Precision	Recall	F1-score
Week 1	0.7826	0.7647	0.9286	0.8387
Week 2	0.6700	0.7500	0.5660	0.6452
Week 3	0.7913	0.7895	0.7895	0.7895

Table III shows the 70-30 split, there was a slight decrease in overall performance compared to the 80-20 split, but the model still showed robust results. In Week 1, the model maintained a high F1-score of 0.7917 and a very strong recall of 0.9500, although precision declined slightly. Week 2 and Week 3 produced relatively balanced values for all metrics, indicating that the model was still capable of learning effectively with 70% of the data, though it benefitted from more training data in the 80-20 case.

Table III. Performance Metrics for 70-30 Train-Test Split

Train-Test Split 70-30	Accuracy	Precision	Recall	F1-score
Week 1	0.7143	0.6786	0.9500	0.7917
Week 2	0.6711	0.7183	0.6375	0.6755
Week 3	0.7238	0.7176	0.7219	0.7198

As seen in Table IV, the performance began to degrade more noticeably with the 60-40 split. In Week 1, the recall remained high at 0.9231, but both precision and accuracy dropped, affecting the overall F1-score. Week 2 showed the weakest results under this split, with all four metrics dipping significantly. This suggests that with only 60% of the data for training, the model's ability to generalize decreased. Week 3, while slightly better, still did not reach the performance levels observed with the higher training ratios.

Table IV. Performance Metrics for 60-40 Train-Test Split

Train-Test Split 60-40	Accura-cy	Precision	Recall	F1-score
Week 1	0.6739	0.6486	0.9231	0.7619
Week 2	0.6131	0.6304	0.5743	0.6010
Week 3	0.7211	0.7124	0.7188	0.7156

In the 50-50 split, performance showed a mixed trend. Week 1 surprisingly still showed strong recall and an F1-score close to that of higher split ratios, but as with other splits, Week 2 again showed weaker performance in all metrics. Week 3 showed relatively stable values but was not significantly better than the 60-40 split. This indicates that while the model can function with 50% training data, the trade-off in performance becomes more apparent.

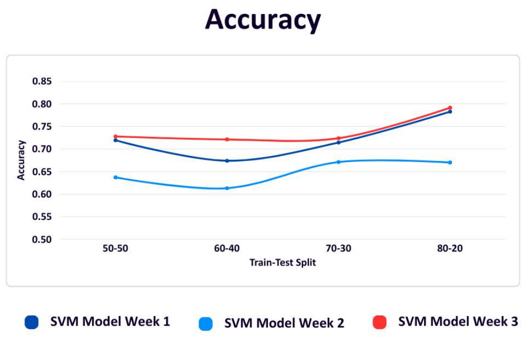

Fig. 2 Accuracy Graph

The Fig. 2 graph compares the accuracy of SVM models trained over three weeks using different train-test splits (50-50 to 80-20). Week 3's model, trained on the largest dataset, shows the highest and most consistent accuracy, peaking at 0.7913 with an 80-20 split. Week 1's model, despite using the smallest dataset, performs well with high recall and reaches 0.7826 accuracy, suggesting good generalization. In contrast, Week 2's model has the lowest and flattest accuracy curve, likely due to an imbalanced or less effective dataset. Overall, more data and balanced classes improved model performance over time.

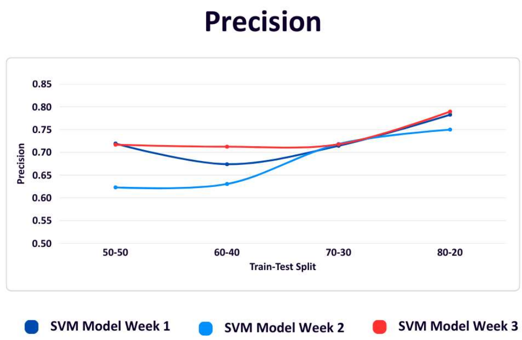

Fig. 3 Precision Graph

The Fig. 3 Precision graph shows that the SVM model's precision improves as the training data increases (toward the 80-20 split) for all three weeks. Week 3 (red) performs best overall due to having the largest and most balanced dataset. Week 1 (dark blue), despite limited data, maintains decent precision. Week 2 (light blue) starts lower but improves steadily with more training data. This highlights that more and balanced data leads to more accurate predictions of fake news.

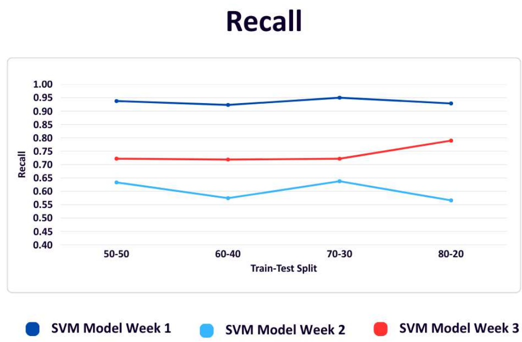

Fig. 4 Recall Graph

The Recall graph shows how well the SVM model correctly identifies fake news across different train-test splits over three weeks. Week 1 (dark blue) consistently has very high recall- above 0.92, due to a smaller dataset and possibly overfitting. Week 3 (red), with more balanced and larger data, shows a steady increase in recall, peaking at 0.7895 in the 80-20 split, indicating better real-world performance. Week 2 (light blue) fluctuates more and has lower recall, reflecting challenges from limited and less balanced data. Overall, recall improves with more training data, especially in Week 3.

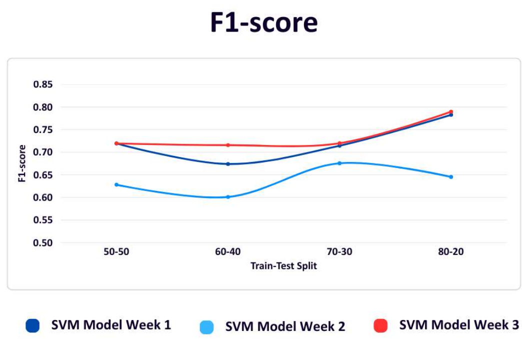

Fig. 5 F1-score Graph

The F1-score graph summarizes the balance between precision and recall for the SVM model across different weeks and train-test splits. In Week 1 (dark blue), performance improves from 0.67 to 0.78 as the training data increases. Week 2 (light blue) has the lowest F1-scores overall, ranging from 0.60 to 0.675, due to lower recall and precision. Week 3 (red) maintains a stable F1-score early on and shows the best result (0.7857) in the 80-20 split, reflecting strong generalization as training data increases. Overall, the F1-score improves with larger training sets, and Week 3 shows the most balanced and robust performance.

In conclusion, the results clearly show that the model performs best when trained on a larger portion of the dataset. The 80-20 and 70-30 splits yielded the most favorable and consistent results, especially in terms of recall and F1-score. As the training data decreased, performance dropped, particularly in Week 2 across all splits. This highlights the importance of using sufficient training data for achieving reliable and generalizable machine learning outcomes. Overall, the study emphasizes that data quantity and potentially data quality significantly influence model success over time.

V. FUTURE WORK

The current system provides a strong foundation for further enhancements aimed at improving the adaptability and effectiveness of fake news detection. One potential direction is the automation of model selection, wherein the optimal algorithm can be dynamically chosen based on comprehensive performance metrics such as precision, recall, and F1-score, rather than accuracy alone. Furthermore, expanding the scope of web scraping to incorporate diverse data sources, multilingual content, and multimedia formats, such as images and videos, can substantially enrich the training dataset and enhance the model's generalization capabilities. The integration of advanced deep learning architectures, including Long Short-Term Memory (LSTM) networks and transformer-based models like BERT, may also contribute to improved contextual understanding and classification accuracy.

VI. CONCLUSION

In this paper, we developed a fake news detection system using machine learning techniques such as Logistic Regression, Support Vector Machine (SVM), and Naive Bayes. The system evaluates news article text to determine its authenticity. To ensure ongoing relevance and accuracy, we implemented dynamic model updating by selecting the best performing algorithm based on accuracy evaluations. The model is retrained using newly collected data obtained through web scraping, allowing it to adapt to emerging trends and linguistic patterns in fake news. This dynamic approach significantly enhances the model's reliability and robustness over time, making it a practical solution for combating the evolving challenge of misinformation.

REFERENCES

[1] W. Shahid et al., "Detecting and Mitigating the Dissemination of Fake News: Challenges and Future Research Opportunities," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 4649-4662, Aug. 2024

[2] A. H. J. Almarashy, M. -R. Feizi-Derakhshi and P. Salehpour, "Enhancing Fake News Detection by Multi-Feature Classification," in IEEE Access, vol. 11, pp. 139601-139613, 2023

[3] M. Babar, A. Ahmad, M. U. Tariq and S. Kaleem, "Real-Time Fake News Detection Using Big Data Analytics and Deep Neural Network," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 5189-5198, Aug. 2024

[4] S. A. Aljawarneh and S. A. Swedat, "Fake News Detection Using Enhanced BERT," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 4843-4850, Aug. 2024

[5] J. Wang, S. Makowski, A. Cieślik, H. Lv and Z. Lv, "Fake News in Virtual Community, Virtual Society, and Metaverse: A Survey," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 4828-4842, Aug. 2024

[6] A. H. J. Almarashy, M. -R. Feizi-Derakhshi and P. Salehpour, "Enhancing Fake News Detection by Multi-Feature Classification," in IEEE Access, vol. 11, pp. 139601-139613, 2023

[7] G. G. Devarajan, S. M. Nagarajan, S. I. Amanullah, S. A. S. A. Mary and A. K. Bashir, "AI-Assisted Deep NLP-Based Approach for Prediction of Fake News From Social Media Users," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 4975-634985, Aug. 2024

[8] A. Altheneyan and A. Alhadlaq, "Big Data ML-Based Fake News Detection Using Distributed Learning," in IEEE Access, vol. 11, pp. 29447-29463, 2023

[9] M. A. Wani et al., "Toxic Fake News Detection and Classification for Combating COVID-19 Misinformation," in IEEE Transactions on Computational Social Systems, vol. 11, no. 4, pp. 5101-5118, Aug. 2024
[10] V. Asha, A. P. Nirmala, A. A. Raj, A. Yadav, A. Sen and A. Rajan, "An Effective Assessment of Machine Learning Approaches for Fake News Detection," 2024 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2024
[11] T. Mahmud, I. Hasan, M. T. Aziz, T. Rahman, M. S. Hossain and K. Andersson, "Enhanced Fake News Detection through the Fusion of Deep Learning and Repeat Vector Representations," 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2024
[12] M. Luqman, M. Faheem, W. Y. Ramay, M. K. Saeed and M. B. Ahmad, "Utilizing Ensemble Learning for Detecting Multi-Modal Fake News," in IEEE Access, vol. 12, pp. 15037-15049, 2024
[13] M. Luqman, M. Faheem, W. Y. Ramay, M. K. Saeed and M. B. Ahmad, "Utilizing Ensemble Learning for Detecting Multi-Modal Fake News," in IEEE Access, vol. 12, pp. 15037-15049, 2024
[14] E. Hashmi, S. Y. Yayilgan, M. M. Yamin, S. Ali and M. Abomhara, "Advancing Fake News Detection: Hybrid Deep Learning With FastText and Explainable AI," in IEEE Access, vol. 12, pp. 44462-44480, 2024
[15] Cat deep system for intrusion detection in software defined networking,International Journal of Intelligent Information and Database Systems