OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 03 - Issue 05 (2026)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

CONSISTENCY-BASED EVALUATION OF SHAP AND LIME EXPLANATIONS FOR MACHINE LEARNING-BASED FAKE NEWS DETECTION

Hira Junejo[1], Noor Ahmed Shaikh[2], Riaz Ahmed Shaikh[3], Hina Kareem[4], Manahil Shaikh[5]

Institute of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan.

***Corresponding Author:** (hirajunejo505@gmail.com)*

***DOI:**(https://doi.org/10.71146/kjmr937)*

Article Info

Abstract

The detection of fake news is a crucial research issue since false and misleading information can easily proliferate in online news and social media. While many machine learning models can perform well in classification, it is hard to trust their predictions when there is no clear and reliable explanation. This paper proposes a consistency-based evaluation for SHAP and LIME explanations for fake news detection using machine learning. The study is based on the WEL Fake dataset, which contains fake news (label 0) and real news (label 1). The final dataset after preprocessing has 63,547 articles, 34,788 fake news and 28,759 real news articles. A machine learning pipeline is created based on TF-IDF features, model selection and validation-based tuning of thresholds for unigram textual features. A balanced Logistic Regression classifier with 80,000 TF-IDF features and a tuned threshold of 0.52 is the best model. On the test set, the model achieves 96.06% accuracy, 95.65% F1-score, 99.31% ROC-AUC, 99.18% PR-AUC, and 92.05% Matthews correlation coefficient. The main contributions of this work are not only fake news classification but also the corrected comparison between SHAP and LIME explanations. For a fair comparison between SHAP and LIME, these both are aligned to the same predicted class, and the explanation features are normalized during the experiment at word level before performing evaluation. Top-K overlap ratio, Jaccard similarity, Spearman correlation, Kendall correlation and sign agreement are used to measure the explanation consistency. The overlap ratio, Jaccard similarity, Spearman correlation, Kendall correlation and sign agreement of SHAP and LIME at Top-10 are 76.94%, 64.35%, 80.42%, 70.32% and 98.25%, respectively. LIME repeated-run stability is also measured, and it has a Jaccard stability of 70.62% and a Spearman stability of 88.52%. The faithfulness deletion test also reveals that the average drop in model confidence is 13.94% when the Top-10 SHAP-selected words are deleted, and 12.93% when the LIME-selected words are deleted. The results indicate that SHAP and LIME can offer consistent and meaningful explanations for fake news detection with appropriate class alignment and feature normalization.

Keywords: *Explainable artificial intelligence; SHAP; LIME; Explanation consistency; Machine learning; Explanation stability; Faithfulness analysis.*

1. Introduction

The speedy growth of online news and social media has transformed how people generate, share and consume information. Now, news sometimes can be disseminated through digital media platforms in less than a matter of seconds and can reach many people before it is confirmed. This has made information more accessible, but has also led to the proliferation of fake news, misinformation and misleading narratives. Fake news can shape public opinion, erode trust in news media, impact on democratic processes, and cause confusion during public health, political and social events [1,2,3]. Hence, automatic fake news detection is an important research problem in the field of natural language processing and machine learning. Fake news detection is frequently considered as a supervised text classification task. In this case, a model is trained with labelled news articles and tested to see if an unseen article is fake or real. Several benchmark datasets have been used for this research area, such as LIAR, Fake Newsnet, and WELFake [4] [5] [6]. These datasets are structured and used in different ways. LIAR is a short political statement dataset, FakeNewsNet is a news content dataset along with social context, and WELFake is article-level textual dataset along with the titles, body text, and binary labels [4] [5] [6]. The WELFake dataset is used as the primary experimental dataset in this study, as it is an article-level fake news dataset. In this data set, the class 0 indicates fake news and class 1 indicates real news [6]. In recent years, numerous studies have demonstrated high accuracy in fake news detection with the help of machine learning, deep learning, ensemble learning, and transformer-based models [7], [8], [9]. But, for a sensitive task like fake news detection, the predictive accuracy is not sufficient. A model can make the right decision to classify a news article, but if it does not explain why the article was classified as fake or real, users may not trust the decision. This is particularly relevant as fake news detection can impact public opinion, political awareness, and social trust. Therefore, explainability is becoming a crucial feature in fake news detection systems [10] [11].

Explainable Artificial Intelligence (XAI) is a field of AI that seeks to make machine learning predictions more understandable. Explanation methods can be used to reveal the words or textual features that contributed to a model's decision in text classification. This can help researchers and users to see if the model is using meaningful evidence or weak shortcuts such as repeated source names, writing style, or patterns in the data set [12, 13]. This is especially helpful in the case of fake news detection, where a model should not just give a label, but also give a reasonable explanation to support the decision. LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations) are two popular post-hoc explanation techniques. LIME creates a simple local model around the prediction by perturbing the input and explains the prediction by fitting the local model to the perturbed inputs [14]. SHAP uses Shapley values from cooperative game theory [15] to assign contribution scores to features. Both methods are popular, as they can be used to explain trained models without the need for the model architecture to be interpretable. In fake news detection, they are frequently employed to show key words that affected a fake or real prediction [16] [17].

But, if you just want to show example explanations, it is not enough to use SHAP or LIME. One explanation may be aesthetically pleasing, but it does not necessarily mean that the explanation is reliable. SHAP and LIME may point to very different words for the same prediction, leading to conflicting evidence for the user. Likewise, if LIME gives different explanations on multiple runs for the same article, then LIME's explanation may not be stable. Hence, the quality of the explanation should be measured by measurable criteria like agreement, ranking similarity, stability and faithfulness [18,19,20]. This study

aims to solve this problem by assessing the consistency of SHAP and LIME explanations for machine learning-based fake news detection. This work does not consider explainability as a visual add-on, but rather asks the question: Do two popular explanation methods agree when explaining the same model prediction? TF-IDF (Term-Frequency-Inverse Document Frequency) based text features and a machine learning classifier trained on the WELFake dataset are used in the study. SHAP and LIME are then used to explain the model's predictions, and multiple consistency measures are used to compare the explanations.

One of the methodological issues in SHAP-LIME comparison is that both SHAP and LIME should explain the same prediction target. SHAP values can naturally explain the direction of one class, and LIME can explain the predicted class of the model in binary classification. This direction is not aligned; the comparison can become unfair and misleading. To solve this problem, the implementation is done to make both SHAP and LIME explanations align with the predicted class. This guarantees that both methods are tested in the same decision direction.

Another critical issue is that the feature level comparability**.** LIME typically uses word-level features for text predictions, while TF-IDF models might generate features in a different format based on preprocessing. For the sake of comparison, in this study, the unigram-level TF-IDF features are used and the explanation features are normalized before agreement is calculated. This involves processing such as lowercasing, punctuation removal, space normalization, and other features that form unification. These steps are used to render SHAP and LIME explanations and are more directly comparable.

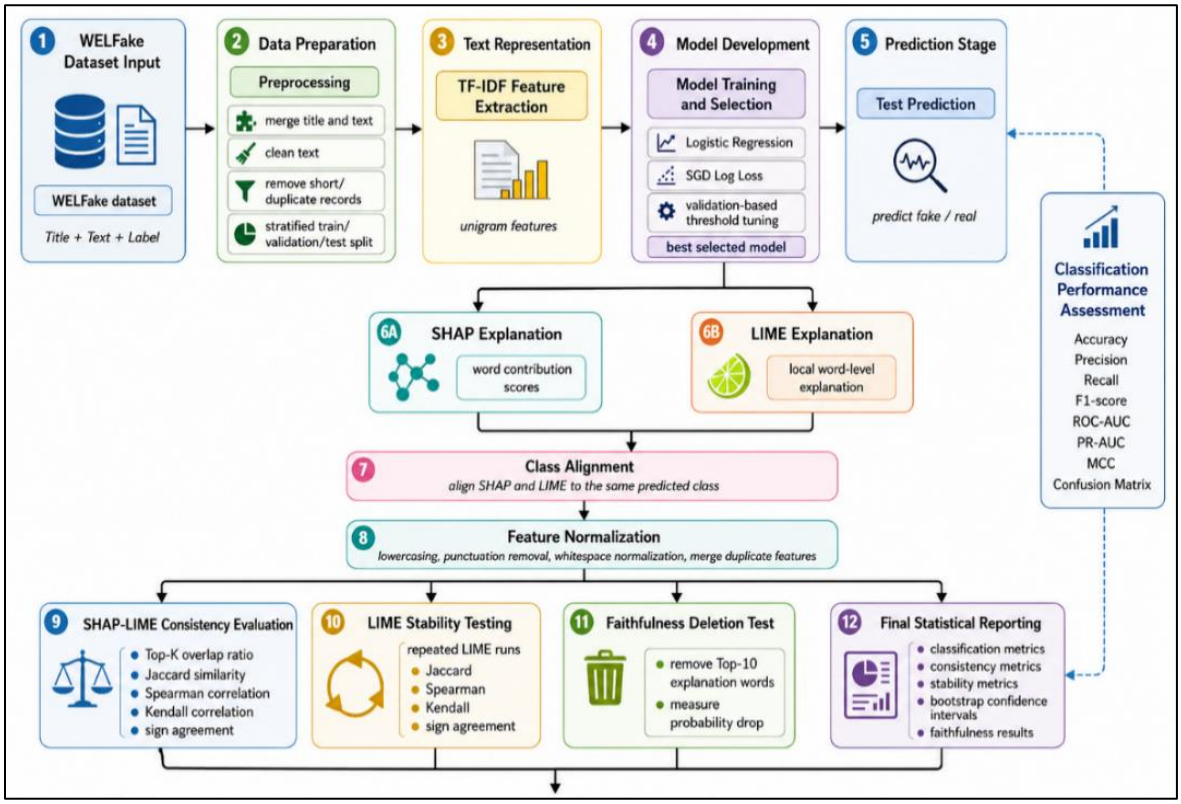


Fig. 1. Proposed framework for consistency-based evaluation of SHAP and LIME explanations in fake news detection.

The overall framework of the proposed consistency-based SHAP-LIME evaluation system for fake news detection is shown in Fig. 1. The WELFake dataset is the starting point, with cleaning, duplicate removal and stratified splitting of the title and text fields. The cleaned textual data is then converted to TF-IDF features and fed to machine learning models for fake and real news classification. Following the prediction, the chosen model is explained by both SHAP and LIME to find the important word-level contributions. For a fair comparison, the explanations are aligned to the same predicted class and normalized before being evaluated. It then quantifies SHAP-LIME consistency, LIME stability and explanation faithfulness using deletion-based testing. Finally, the system presents classification performance and explanation reliability metrics, which offer both predictive and interpretability based evaluation of the proposed fake news detection approach.

This paper makes the following main contributions. First, the study proposes a full machine learning pipeline for fake news detection based on the WELFake dataset and textual features with TF-IDF. Second, it proposes a corrected SHAP-LIME comparison process, which is aligned to the predicted class and normalized word-level features. Third, it assesses the reliability of explanations with several metrics, such as Top-K overlap, Jaccard similarity, Spearman correlation, Kendall correlation, sign agreement, LIME repeated-run stability, bootstrap confidence intervals and faithfulness deletion testing. These contributions are significant because fake news detection systems need to be not only accurate, but also explainable, stable and trustworthy.

2. Related Work

The detection of fake news is a research field of great importance since false and misleading information can spread very quickly in the online world. This information can impact public opinion, social trust, political understanding, and public response during sensitive events. Previous studies primarily concentrated on detecting fake news based on textual information, and subsequent studies extended the problem to incorporate social context, user behavior, propagation patterns, and multimodal information. The key research fields related to the current study are summarized as follows: fake news detection datasets, fake news classification using machine learning, explainable artificial intelligence for fake news detection, explanation methods (SHAP and LIME), and the necessity of consistency evaluation of explanations.

2.1 Fake News Detection Datasets

The quality of fake news detection model is highly dependent on the dataset used for training and testing. A number of benchmark datasets have been created for this purpose. The LIAR dataset is one of the earlier benchmark datasets and consists of short political statements with different truthfulness labels [20]. It can be helpful for fact-checking research, but is not the same as fake news detection at the article level, which is a short statement. FakeNewsNet is another important dataset that includes news content, social context, and dynamic information for studying fake news on social media [17]. It can be used in fake news research in general as it offers both content-based and social-context information.

The WELFake dataset is a fake news dataset that combines several different sources of fake news data, such as Kaggle, McIntire, Reuters, and BuzzFeed Political data [15] [16] at the article level. It includes news titles, news text and binary labels (0 for fake news and 1 for real news) [15]. In the present study, the

research is focused on article level text classification and explanation analysis, so WELFake is suitable for this research. It contains sufficient textual content for TF-IDF feature extraction and word level explanation comparison with SHAP and LIME. The selection of the dataset is still a crucial problem in fake news detection. Some data sets may be subject-specific, some may be politically oriented, and some may contain source-specific writing styles. These characteristics may impact classification performance and explanation quality. Therefore, it is important to explicitly state the source of the dataset, the meaning of the labels, the class distribution, the preprocessing steps and the splitting strategy in fake news detection research. Dataset surveys and benchmarks have also pointed out that the lack of uniformity in using datasets can hinder the fair comparison of fake news detection studies [2, 3].

2.2 Machine Learning-Based Fake News Detection

The traditional machine learning methods are widely used for fake news detection due to their efficiency, interpretability, and effectiveness in text classification. The most popular methods are TF-IDF, bag-of-words, and linguistic features combined with classifiers like Logistic Regression, Support Vector Machines, Naive Bayes, Random Forest, and boosting. Fake news detection methods have been reviewed and are still applicable, particularly if there is a significant difference in the textual content of fake news and real news in the dataset [11] [21]. Fake news detection has also been performed using deep learning techniques. Deeper semantic and contextual representations can be learned from news text using recurrent neural networks, convolutional neural networks, LSTM-based models, and transformer-based models. Padalko et al. [12] suggested a fake news classification method using LSTM, and Tian et al. [18] compared the machine learning and deep learning methods for fake news detection. These studies show that deep models can be effective, but they are more costly and less interpretable than traditional machine learning models. Ensemble learning and more complex architectures for fake news detection are also recent research interests.

Kukkar and Kaur [9] introduced an adaptive ensemble classifier with LIME and SHAP based interpretability. Jadhav et al. [7] highlighted the explainable multilingual and multimodal fake-news detection, which is going beyond the scope of text classification. Nevertheless, many models remain in need of further development of explanation evaluation, as prediction accuracy does not necessarily imply that the model decisions are explainable or trusted. TF-IDF based machine learning model is used in the present study because the primary goal is not only fake news classification, but also the assessment of the consistency of explanations. TF-IDF features are more directly word-level explanation features because each feature is a textual token. This makes the comparison between SHAP and LIME easier than for models with more complex internal representations, which are less easy to relate to individual words.

2.3 Explainable Artificial Intelligence in Fake News Detection

The importance of Explainable Artificial Intelligence in fake news detection is that the classification decision may impact public trust and credibility. Users, journalists, researchers and decision makers may not be satisfied with a model that only provides “fake” or “real”. They also might need to know what text features affected the prediction. Athira et al. [1] discussed the explainable AI approaches that have been used in fake news detection and emphasized the need for transparency in this field. Gongane et al. [5] also conducted a survey of explainable AI techniques for fake news and hate speech detection in social media.

For fake news detection, explanation methods can highlight which words or features were driving the model towards a fake or real prediction. This allows us to verify that the model is learning evidence from data rather than weak shortcuts like frequent source names, writing styles, political entities, or signals that are specific to the dataset itself. Vimbi et al. [19] examined LIME and SHAP, two widely used methods for explaining AI models, and their significance in various machine learning applications. There are some risks to explainability as well which are later discussed in this study. Kozik et al. [8] showed that XAI can be used to fool a fake news detection method. This means that explanations can expose weaknesses in the model and could be used by attackers to try to evade detection. So explainability needs to be approached with caution. It should facilitate trust and analysis, but researchers also need to assess the reliability, stability and meaningfulness of explanations.

2.4 SHAP and LIME for Text Explanation

Two of the most popular post-hoc explanation techniques are LIME and SHAP. LIME explains individual predictions by perturbing the input and fitting a simple interpretable model around the local prediction [14]. Typically, this involves changing or removing words and seeing how the model prediction changes in the case of text classification. The final LIME explanation shows words that agree or disagree with the prediction. While LIME is useful due to its model-agnostic and easy-to-understand nature, it can be sensitive to random sampling, perturbation settings, and the number of samples generated. SHAP uses Shapley values from cooperative game theory [10] to explain predictions. It gives every feature a contribution score, which indicates the contribution of the feature to the prediction. SHAP is well-founded theoretically and has been adopted in machine learning interpretation. For text classification, SHAP can indicate which words are responsible for the model output. But, SHAP can be expensive depending on the model type and feature space.

Some recent research has been conducted comparing and reviewing SHAP and LIME in various fields. Givisis et al. [4] conducted a comparison between explainable AI models (SHAP and LIME), and Hermosilla et al. [6] performed a comparative study of SHAP and LIME in forensic analysis. These works demonstrate that both approaches are valuable, and that they can generate varying explanations based on the model, data, and explanation settings. Rathod et al. [13] applied explainability in a scalable fake news detection system, while Kukkar and Kaur [9] used LIME and SHAP-based explainability in an adaptive ensemble fake news detection system. But, most of the studies only provide examples of explanations using SHAP and LIME, and do not systematically measure the consistency of the two explanation methods.

2.5 Explanation Consistency and Stability

Explanation consistency is an important issue in explainable fake news detection. If two explanation methods provide very different explanations for the same prediction, the reliability of the explanation is called into question. For instance, if SHAP shows one set of words and LIME shows another set of words, users might not know which explanation to believe. Likewise, if LIME gives different explanations for the same article when run multiple times, the explanation might not be stable enough to be interpreted reliably.

In recent XAI studies, it has been suggested that explanations be assessed based on measurable criteria like agreement, ranking similarity, stability, and faithfulness [4] [19]. Agreement is the extent to which the different methods of explanation emphasize the same features. Ranking similarity is the extent to which the methods rank important features in a similar manner. Stability is the consistency of the explanation method when it is applied to the same data set multiple times or to slightly different data sets. Faithfulness indicates if the highlighted features actually affect the model's prediction. To compare the important words selected by SHAP and LIME, Top-K overlap and Jaccard similarity can be used for text classification. Spearman and Kendall correlations can be used to measure ranking agreement. Sign agreement can indicate if both methods agree that a word supports or opposes the predicted class. Important words can be deleted from the text and the change in model confidence can be used to test faithfulness. These metrics give a better assessment than just a few explanation examples.

The present study goes in this direction by assessing SHAP and LIME explanations with several consistency and stability metrics. This study differs from studies that merely visualize explanations because it measures the strength of the agreement between the two methods after aligning them to the same predicted class and normalizing word-level features. This renders the comparison more reliable and appropriate for the purpose of research reporting.

2.6 Research Gap

Based on the literature surveyed, fake news detection has been extensively studied using machine learning, deep learning, benchmark datasets and explainable AI approaches [1, 2, 3, 11, 21]. It also reveals that SHAP and LIME are popular methods for explaining machine learning predictions [10, 14, 19]. But there are three key gaps.

First, many fake news detection studies focus mainly on the classification performance and explainability is considered as a secondary output. They present accuracy, F1 score or AUC and then display a few explanation examples. This is not a complete assessment of the reliability of the explanations.

Second, SHAP and LIME are frequently applied together, but quantitative comparisons of the explanations are not always made. Ranking similarity, sign agreement and stability are not measures of overlap and without these measures it is hard to determine whether the two explanation methods are providing consistent evidence for the same prediction.

Third, if the methods are not completely matched up correctly, explanation comparison can be misguiding and misleading**.** For binary classification, SHAP can explain the direction of class 1, and LIME can explain the predicted class. This direction is not precise and accurate; the comparison may result in artificially weak consistency throughout. Likewise, SHAP and LIME features may not be normalized to the same word-level representation, which could lead to underestimation of feature overlap.

In this study, we aim to fill these gaps by providing a consistency-based evaluation of SHAP and LIME explanations for fake news detection using machine learning. The proposed approach explains in the direction of the predicted class, normalizes word level features, checks for agreement at multiple Top-K settings, checks for LIME repeated run stability, reports bootstrap confidence intervals, and performs a faithfulness deletion test. This provides a more comprehensive view of the reliability of the explanation in fake news detection.

Table 1. Summary of Related Work and Research Gap

Study	Main focus	Dataset / domain	Explainability method	Main limitation related to this study
Athira et al. [1]	Survey of XAI in fake news detection	Fake news detection studies	XAI methods	Survey-level discussion, not an experimental SHAP-LIME consistency evaluation
D'Ulizia et al. [2]	Fake news dataset survey	Fake news datasets	Not the main focus	Focuses on datasets, not explanation consistency
Galli et al. [3]	Benchmarking fake news detection	Fake news datasets	Not the main focus	Benchmark focus, limited SHAP-LIME consistency analysis
Givisis et al. [4]	Comparison of XAI models	General XAI applications	SHAP and LIME	Not focused specifically on fake news detection
Gongane et al. [5]	XAI for fake news and hate speech	Social media platforms	SHAP, LIME, and other XAI methods	Survey-level discussion, not a corrected experimental consistency pipeline
Shahane [15], [16]	WELFake dataset	Article-level fake news	Not the main focus	Dataset source, not explanation evaluation
Shu et al. [17]	FakeNewsNet dataset	News content and social context	Not the main focus	Dataset repository, not SHAP-LIME consistency
Tian et al. [18]	ML and DL comparison	Fake news detection	Limited XAI focus	Focuses on model performance more than explanation agreement
Wang [20]	LIAR benchmark dataset	Political statements	Not the main focus	Short-statement dataset, not article-level SHAP-LIME analysis
Kukkar and Kaur [9]	Adaptive ensemble fake news detection	Fake news datasets	LIME and SHAP	Uses explainability, but not mainly focused on corrected SHAP-LIME consistency
Kozik et al. [8]	XAI risk in fake news detection	Fake news detection	XAI-based analysis	Shows XAI vulnerability, not explanation agreement as reliability measure
Present study	Consistency-based SHAP-LIME evaluation	WELFake	SHAP, LIME, stability, faithfulness	Directly focuses on corrected quantitative explanation consistency

3. Proposed Methodology

The proposed methodology for consistency-based evaluation of SHAP and LIME explanations in machine learning-based fake news detection is presented here. The methodology aims to assess the predictive accuracy of a fake news classifier as well as the trustworthiness of its explanations. The overall idea is to train a fake news detection model using text data, generate SHAP and LIME explanations for the same predictions, align the explanations to the predicted class, normalize the explanation features and then compare the explanations using consistency, stability and faithfulness measures. The proposed methodology has six main stages: dataset preparation, feature extraction, model training and selection, prediction and test evaluation, explanation generation, and explanation reliability evaluation.

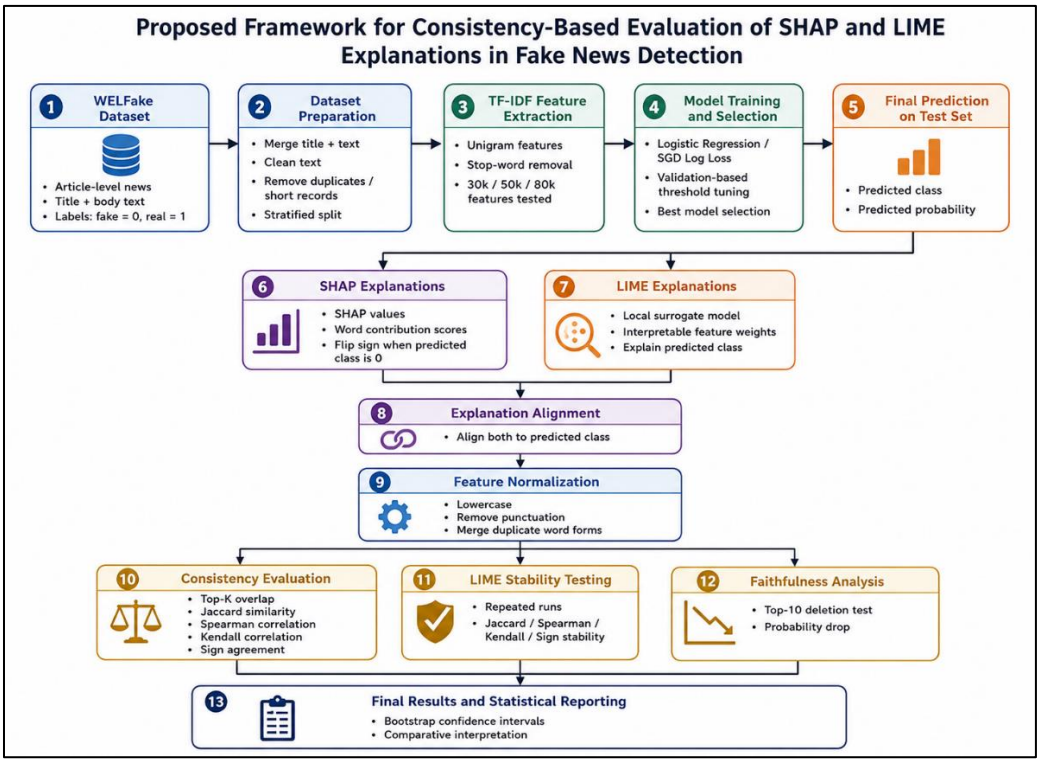


Fig. 2. Overall framework of the proposed consistency-based SHAP-LIME evaluation system for fake news detection.

The process starts with the WELFake dataset which consists of article level news data with title, body text and class label. The title and text are combined, the text is cleaned, duplicate or very short records are removed and the data is split using a stratified split in the dataset preparation stage. The prepared text is then transformed to a numerical form by TF-IDF feature extraction, in which unigram features are created and various feature sizes are evaluated.

These features are utilized in the model training and selection phase, where machine learning models like Logistic Regression and SGD Log Loss are trained, threshold tuning is done on the validation set, and the best model is selected. The selected model is then used to make the final prediction on the test set, which consists of the predicted class and the probability of the class. Following prediction, both SHAP and LIME are used to provide explanations. SHAP values are used to calculate word contribution scores and LIME values are used to calculate local interpretable feature weights for the predicted class. In order to compare both explanation methods fairly, the next step is explanation alignment, which is to align both to the same predicted class. This is then followed by feature normalization, which mainly involves processing like lowercasing, removing punctuation marks, and removing duplicate word forms from the explanation terms.

The framework then carries out three different key evaluation functions as soon as it has been aligned and normalized. Consistency evaluation is done in the first step, where SHAP and LIME are compared using Top-K overlap, Jaccard similarity, Spearman correlation, Kendall correlation and sign agreement. Second, LIME stability testing determines if LIME provides the same explanations in multiple runs. Third, faithfulness analysis is a measure of the quality of the explanation, where the Top-10 important words are removed and the drop in probability is measured. Finally, all results are summarized in the final results

and statistical reporting stage, which includes bootstrap confidence intervals and comparative interpretation of the results.

3.1 Dataset Preparation

The WELFake dataset is used as the main dataset in this study. It is an article-level fake news data set with news titles, news text and binary labels [15, 16]. This data is appropriate for this study as the proposed method is to assess word-level explanations in news articles. WELFake is a longer article text data set compared to short-statement data sets like LIAR [20] which enables more meaningful explanation analysis and richer feature extraction.

The original WELFake data set is comprised of 72,134 records. The title and text are combined in the preprocessing phase to provide a single textual input for each article. This is helpful because the title of the article usually has a lot of clues as to what the article is about, and the body of the article has more detailed contextual clues. Text cleaning is performed after merging, which includes removing empty or very short records, normalizing spaces, and removing duplicate text entries. The final cleaned data set has 63,547 articles.

The data set is binary, with fake news being labeled as 0 and real news being labeled as 1 [15]. The cleaned data set has 34,788 fake articles and 28,759 real articles. Stratified splitting is used to split the data into training, validation, and test sets. Stratification is necessary to ensure that the class distribution is maintained in each split and that one class is not.

Table 1. Dataset distribution after preprocessing and stratified splitting

Dataset stage / split	Fake news, label 0	Real news, label 1	Total samples	Fake %	Real %
Raw WELFake dataset	—	—	72,134	—	—
Cleaned dataset	34,788	28,759	63,547	54.74%	45.26%
Training set	24,351	20,131	44,482	54.74%	45.26%
Validation set	3,479	2,876	6,355	54.74%	45.26%
Test set	6,958	5,752	12,710	54.74%	45.26%

As seen in Table 1, the class distribution is slightly imbalanced with fake news making up 54.74% of the cleaned data set and real news making up 45.26%. The ratio is the same for the training, validation and test sets, indicating that stratified splitting was done correctly. This is crucial because the explanation results should be obtained from a model trained and tested with a fixed class distribution.

3.2 Text Feature Extraction

Once the data is prepared, the cleaned article text is transformed to numerical features using TF-IDF. TF-IDF is widely used in text classification since it is a measure of the importance of a word in a document with respect to the entire collection. Words that are used often in one article, but not often in other articles, are given higher importance.

Unigram level TF-IDF features are used in this study. This implies that each feature is a single word, not a phrase or n-gram. The use of unigrams is for methodological fairness. Word level text prediction is a common use of LIME [14]. If SHAP explanations are based on unigram and bigram TF-IDF features, and LIME explanations are primarily word-level, then the comparison may be unfair. For instance, SHAP can point to a phrase like "white house", whereas LIME can point to each individual word "white" and "house". This would minimize the overlap of features, even if both are referring to the same textual evidence. To make SHAP and LIME more directly comparable, unigram features are used. The TF-IDF vectorizer is set up to lower case, remove English stop words, scale term frequencies sub linearly and normalize using L2 normalization. Model selection is performed with different maximum feature sizes such as 30,000, 50,000 and 80,000 features. The final selected model is with 80,000 TF-IDF features as this configuration gives the best validation F1-score.

3.3 Model Training and Selection

The proposed pipeline uses multiple machine learning models and tests them on the training and validation sets. Candidates are balanced Logistic Regression and SGD-based logistic-loss classifiers. These models are appropriate for this study because they can be efficiently applied to high dimensional text data and can be used with post-hoc explanation approaches. Logistic Regression is particularly appropriate for this research as it is relatively interpretable and has good text classification performance. A linear model with TF-IDF features offers a transparent connection between words and model decisions, which is the objective of this work, as well as fake news classification. This allows for easier analysis of whether SHAP and LIME identify similar word-level evidence.

The model selection is done on the validation set. The validation probabilities are calculated at various decision thresholds for each model and feature-size setting. The final threshold is chosen according to the validation F1 score and balanced accuracy (when necessary). Threshold tuning is crucial as the default threshold of 0.50 might not always yield the optimal precision-recall trade-off, particularly in the case of a slightly imbalanced dataset.

Table 2. Top validation results from model selection

Rank	Model	Features	Threshold	Validation accuracy	Validation F1	Validation ROC-AUC	Validation PR-AUC	Validation MCC
1	Logistic Regression, C=4, balanced	80,000	52%	96.40%	96.02%	99.35%	99.27%	92.73%
2	SGD Log Loss, alpha=1e-6, balanced	80,000	53%	96.38%	96.00%	99.35%	99.28%	92.70%
3	Logistic Regression, C=4, balanced	50,000	55%	96.38%	95.98%	99.35%	99.27%	92.69%
4	Logistic Regression, C=4, balanced	30,000	52%	96.37%	95.98%	99.34%	99.26%	92.66%
5	SGD Log Loss, alpha=1e-5, balanced	30,000	49%	96.30%	95.93%	99.30%	99.21%	92.54%

The optimal validation result is obtained from the balanced Logistic Regression model with 80,000 TF-IDF features and a threshold of 0.52 as shown in Table 2. Despite the second ranked SGD model having slightly better ROC-AUC and PR-AUC, the Logistic Regression model is chosen since it has the highest validation F1-score and is easier to interpret for word-level explanation analysis.

3.4 Test Prediction and Classification Evaluation

Once the best model and threshold is selected, the final classifier is tested on the held-out test set. The test set is not used for training or model selection. This guarantees that the test performance reported is representative of the model's ability to generalize to unseen news articles.

The classification performance is measured by accuracy, balanced accuracy, precision, recall, F1-score, ROC-AUC, PR-AUC and Matthews correlation coefficient. Accuracy is the percentage of correct predictions. Balanced accuracy is added due to the slight imbalanced nature of the dataset. Precision is the ratio of the number of articles correctly predicted as real to the number of articles predicted as real, and recall is the ratio of the number of articles correctly predicted as real to the number of real articles. F1-score is a balance between precision and recall**.** ROC-AUC and PR-AUC are based on probability for class separation. MCC is also included because it takes into account both the true positives and the true negatives as well as the false positives and false negatives, which makes it a useful metric for binary classification.

The confusion matrix is also explored for class level errors. This is crucial as a fake news detection system should be judged not only in terms of its overall accuracy, but also by the nature of its errors.

3.5 Explanation Generation Using SHAP and LIME

Once the chosen model has made predictions on the test set, SHAP and LIME are used to explain the predictions. The explanation sample set consists of 792 test samples that were chosen from the test predictions. The sample contains both correct and incorrect articles. It is helpful to include misclassified samples so that the analysis can consider explanations not just for successful predictions, but also for model errors.

LIME leverages local explanations by altering the input text and seeing how the model prediction changes [14]. In this study, LIME is set up to provide explanations for the predicted class at the word level. For each LIME explanation, you will be provided with a list of important words and contribution scores. Positive contribution scores mean that the score is in favor of the predicted class, and negative scores mean that the score is against the predicted class. SHAP creates feature contribution scores that are based on Shapley-value explanation principles [10]**.** SHAP gives word-level contribution values for the linear TF-IDF model. These values show the contribution of each feature to the model output. In binary classification, however, SHAP values can naturally point towards class 1. This is a problem when the model predicts class 0.

Both explanations need to explain the same target in order to be comparable with SHAP and LIME. Hence, SHAP values are aligned with the predicted class. When the model is predicting class 1, the SHAP direction remains the same. The SHAP contribution signs are inverted when the model classifies the class

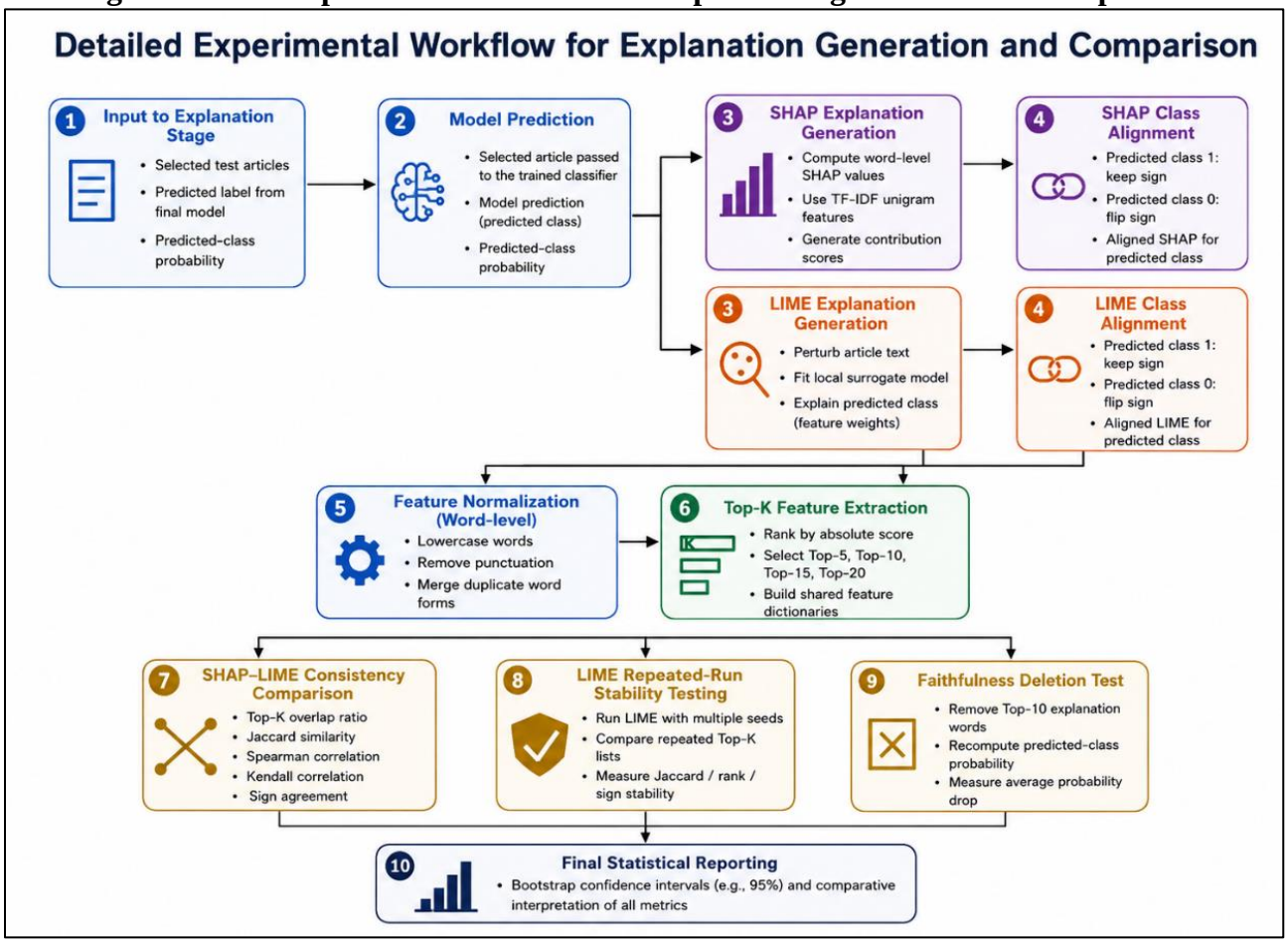
as 0. This means that SHAP explains the actual class predicted by the model and not always class 1. This is necessary to make SHAP comparable to LIME.

3.6 Feature Normalization for Fair Explanation Comparison

After normalizing the features, SHAP and LIME explanations are compared. This is required since SHAP features are from the TF-IDF vectorizer, while LIME features are from perturbed text tokens. If not normalized, the same word can be represented in multiple forms and be considered as multiple features. For instance, if normalization is not used, “Reuters,” “reuters,” and “reuters.” can be considered three distinct words.

Normalization involves lowercasing, removing punctuation, normalizing whitespaces, and removing duplicate normalized features. This increases the reliability of the explanation comparison. SHAP and LIME feature dictionaries are sorted by absolute contribution score after normalization and Top-K features are chosen for comparison. Top-5, Top-10, Top-15 and Top-20 explanation features are evaluated.

Fig. 3. Detailed experimental workflow for explanation generation and comparison.



The experimental workflow shown in fig .3 is for creating and comparing SHAP and LIME explanations is shown**.** The same predictions (class and probability of selected test articles) are sent to SHAP and

LIME. SHAP generates word-level contribution scores and LIME generates local feature weights, based on a surrogate explanation model. Both explanations are then mapped to the predicted class and word-level normalized before extracting Top-K important features. Consistency metrics, LIME repeated-run stability metrics, and faithfulness deletion testing are used to compare these Top-K features. The last stage mainly represents the results with statistical confidence intervals along with the comparative interpretation.

3.7 SHAP-LIME Consistency Metrics

The primary objective of this work is to quantify the consistency of the explanations given by SHAP and LIME for the same fake news prediction. This is done using a number of complementary metrics:

3.7.1 Top-K Overlap Ratio

Top-K overlap ratio indicates the number of common features between SHAP and LIME in the Top-K features. It is computed as:

$$\text{Overlap Ratio} = \frac{|\ \text{SK} \cap \text{LK}\ |}{K}$$

where **SK** is the set of Top-K SHAP features, **LK** is the set of Top-K LIME features, where **K** is the number of selected explanation features. The more overlap indicates the more important words that both methods find.

3.7.2 Jaccard Similarity

The set-level agreement between SHAP and LIME features is measured using Jaccard similarity. It is determined by:

$$\text{Jaccard Similarity} = \frac{|\ \text{SK} \cup \text{LK}\ |}{|\ \text{SK} \cap \text{LK}\ |}$$

This is a more stringent measure than overlap ratio since it takes into account the size of the union of both feature sets. The closer the Jaccard score is to 1, the more similar the feature-sets are.

3.7.3 Spearman Rank Correlation

Spearman correlation is used to check if the order of the explanation features ranked by SHAP and LIME is similar. It is quite helpful because two methods could choose the same features but yet rank them completely different. Spearman correlation is computed on the union of SHAP and LIME features, where missing features are given a score of zero. The higher the Spearman value, the more agreement in the rankings.

3.7.4 Kendall Rank Correlation

Another rank based measure of ordinal agreement between two ranked lists of features is Kendall correlation. It is more stringent than Spearman correlation as it takes into account the agreement in the pairwise ordering. In this study, Kendall correlation is employed in addition to Spearman correlation to give a more robust perspective of ranking consistency.

3.7.5 Sign Agreement

Sign agreement is the agreement between SHAP and LIME on the direction of a feature's contribution. A feature can be in favor of or against the predicted class. Only features common to both methods are used to calculate sign agreement. A high sign agreement suggests that when both the methods identify the same word, they both tend to agree on whether the word supports or weakens the stated prediction.

3.8 LIME Stability Evaluation

Since LIME produces explanations by sampling variations of the input text, it can be sensitive to random perturbations [14]. Hence, the stability of LIME repeated runs is also assessed in this study. For each explanation sample, LIME is run multiple times using different random seeds. Jaccard similarity, Spearman correlation, Kendall correlation and sign agreement are used for comparing the Top-K features from repeated runs.

This analysis indicates if LIME provides the same explanations for the same article when it is run multiple times. If LIME explanations differ significantly, then a good SHAP-LIME agreement score is not sufficient. Thus, stability testing is crucial to assess the trustworthiness of LIME explanations in fake news detection.

3.9 Faithfulness Deletion Test

Consistency and stability are measures of agreement, but do not necessarily indicate that the explanation features are meaningful to the model. Hence, a faithfulness deletion test is also conducted in this study. In this test, the Top-10 words identified by SHAP and LIME are deleted from the input text and the probability of the model's predicted class is calculated again.

If the removal of the selected words decreases the model's confidence, then the explanation is more faithful, because the highlighted words actually affected the prediction. The probability drop is:

$$\text{Probability Drop} = P_{\text{original}} - P_{\text{removed}}$$

where P_{original} is the original predicted-class probability, and P_{removed} is the predicted-class probability after removal of the explanation words.

This test is helpful but needs to be interpreted with care. Deleting words from text can result in unnatural sentences and can sometimes lead to an increase in model confidence rather than a decrease. Hence, the deletion test is employed as a supplementary measure of the quality of explanation, rather than the sole measure.

3.10 Bootstrap Confidence Intervals

To improve the reliability of the reported consistency results, bootstrap confidence intervals are calculated. The explanation results are resampled many times with replacement and the mean value of each metric is computed. The 2.5th and 97.5th percentiles of the bootstrap distribution are used as the 95% confidence interval.

Bootstrap confidence intervals can be used to demonstrate the stability of the consistency results across the explanation sample. A smaller interval means that the metric estimate is more reliable, and a larger interval means that the estimate is less reliable.

3.11 Summary of the Proposed Method

The proposed methodology assesses fake news detection from a performance and explanation point of view. A TF-IDF based classifier is trained and selected based on validation results as the first step. Secondly, the final classifier is tested on a held-out test set. Third, the explanations from SHAP and LIME are produced for the same predictions. Fourth, both explanations are class aligned and word normalized. Consistency metrics, LIME stability, faithfulness deletion and bootstrap confidence intervals are then used to assess the reliability of the explanation.

This methodology directly tackles the primary research objective of this paper: to investigate if SHAP and LIME offer consistent and reliable explanations for fake news detection using machine learning.

4. Experimental Setup

This section presents the experimental setup for the assessment of the proposed fake news detection and explanation consistency framework. It contains the information about the dataset, the pre-processing parameters, the feature extraction parameters, the model selection process, the explanation generation parameters, and the evaluation parameters. This section is intended to enable the experiment to be repeated and to distinguish between the experimental setup and the results.

4.1 Dataset Description

The experiment uses the WELFake dataset as the main benchmark dataset. WELFake is an article-level fake news dataset consisting of news titles, news text, and binary labels [15, 16]. This data set is appropriate for this study as it contains complete articles, not just short statements. This makes it suitable candidate for the fake news classification and word-level analysis of the explanation. The original data set has 72,134 records. Each record consists of a title, article text, and label. The labels in the dataset are 0 for fake news and 1 for real news [15]. In the preprocessing step, the title and the text of the article are combined in one input field. This is because there are useful signals in the title and in the body text that can be used to detect fake news. Empty, very short and duplicate records are removed. The final cleaned dataset has 63,547 articles.

The cleaned data set has 34,788 fake articles and 28,759 real articles. Stratified splitting is used to divide the dataset into training, validation, and test sets. Stratified splitting keeps the fake-real class ratio almost the same in each split. This is crucial to ensure that model training, model selection and final testing are conducted under similar class distributions.

Table 1. Dataset distribution after preprocessing and stratified splitting

Dataset stage / split	Fake news, label 0	Real news, label 1	Total samples	Fake %	Real %
Raw WELFake dataset	—	—	72,134	—	—
Cleaned dataset	34,788	28,759	63,547	54.74%	45.26%
Training set	24,351	20,131	44,482	54.74%	45.26%
Validation set	3,479	2,876	6,355	54.74%	45.26%
Test set	6,958	5,752	12,710	54.74%	45.26%

As can be seen in Table 1, the cleaned data is slightly imbalanced with fake news accounting for 54.74% of the data and real news accounting for 45.26% of the data. All three splits have the same class ratio. This validates that the data split is stable and fair for training, validation and testing.

4.2 Preprocessing Procedure

The preprocessing is kept simple since the aim is to retain readable words for explanation analysis. Because it can make explanation words harder to understand, heavy preprocessing, such as aggressive stemming or deep linguistic filtering is avoided.

The following are the preprocessing steps:

1. Combining the title and body text into a single input field.
2. Deleting records that have no text or very little text.
3. Normalizing spaces and line breaks.
4. Removing duplicate article text.
5. Converting labels to binary numeric format.
6. Partitioning the data into training, validation and test sets.

This is a preprocessing design that can be used for classification and explanation. SHAP and LIME explanations are subsequently interpreted at the word level, so the text should be as close to the original readable form as possible.

4.3 Feature Extraction Settings

TF-IDF is used to transform the cleaned text to numerical features. TF-IDF is chosen due to its popularity in text classification and the fact that it offers a straightforward connection between words and model features. This mapping is crucial for the current study as explanations provided by SHAP and LIME must be compared at the word level. The final implementation is based on unigram TF-IDF features alone. This means that each feature represents a single word. Unigrams are used since LIME typically explains predictions at the word level with perturbations [14]. When bigrams or trigrams are provided, SHAP can choose phrase-level features, and LIME can choose individual words. The comparison might be unfair To make the SHAP and LIME explanations more directly comparable, unigram-level TF-IDF is used.

Table 2. TF-IDF feature extraction configuration

Setting	Value
Feature extraction method	TF-IDF
Token level	Unigram
N-gram range	1,1
Stop words	English
Term-frequency scaling	Sublinear TF
Normalization	L2
Minimum document frequency	2
Maximum document frequency	0.95
Tested feature sizes	30,000; 50,000; 80,000
Final selected feature size	80,000

As seen in Table 2, multiple feature-size settings are tried for model selection. The final selected configuration is 80,000 TF-IDF features as it provides the highest validation F1-score. Unigram features also help to make explanations fair.

4.4 Model Candidates and Selection Strategy

Several linear machine learning models are tested. The models tested are balanced Logistic Regression and logistic-loss classifiers using SGD. These models can be used for high dimensional sparse TF-IDF features and are efficient for large text classification tasks. Balanced models are used due to the slight imbalance in the cleaned dataset. Class balancing is used to mitigate the bias towards the larger class and to facilitate more stable classification between fake and real articles. The validation set is used to select the best model and threshold. The test set is not used for training and is reserved for the final evaluation.

Decision threshold is optimized on the validation set. Instead of using the default threshold of 0.50, the experiment tests thresholds from 0.20 to 0.80. The best threshold is selected based on validation F1-score. When comparing model behavior, balanced accuracy and MCC are also taken into account.

Table 3. Candidate model configuration

Model family	Main configuration	Reason for inclusion
Logistic Regression	Balanced class weights, different C values	Strong linear baseline and interpretable for text features
SGD Log Loss	Balanced class weights, different alpha values	Efficient logistic-loss model for sparse high-dimensional text
TF-IDF feature sizes	30,000; 50,000; 80,000	Tests whether larger vocabulary improves validation performance
Threshold tuning	0.20 to 0.80	Finds better precision-recall balance than default threshold

Table 3 mainly summarizes the candidate model setup**.** The model selection process is tailored to select a classifier that is not only good in terms of performance but also is appropriate for explanation analysis. Logistic Regression is especially appropriate as the contribution of the word-level features is more readily interpretable than in more complex black-box models.

4.5 Final Selected Model

The validation results show that the balanced Logistic Regression model with 80,000 TF-IDF features gives the best validation F1-score. The chosen model has a regularization parameter of C = 4 and an optimized threshold of 0.52.

Table 4. Top validation results from model selection

Rank	Model	Features	Threshold	Validation accuracy	Validation F1	Validation ROC-AUC	Validation PR-AUC	Validation MCC
1	Logistic Regression, C=4, balanced	80,000	52%	96.40%	96.02%	99.35%	99.27%	92.73%
2	SGD Log Loss, alpha=1e-6, balanced	80,000	53%	96.38%	96.00%	99.35%	99.28%	92.70%
3	Logistic Regression, C=4, balanced	50,000	55%	96.38%	95.98%	99.35%	99.27%	92.69%
4	Logistic Regression, C=4, balanced	30,000	52%	96.37%	95.98%	99.34%	99.26%	92.66%
5	SGD Log Loss, alpha=1e-5, balanced	30,000	49%	96.30%	95.93%	99.30%	99.21%	92.54%

Table 4 shows that the selected Logistic Regression model gives the highest validation F1-score of 0.960209. The second ranked model (SGD) has a slightly higher ROC-AUC and PR-AUC, but the Logistic Regression model is selected as the main selection criterion is F1-score, and the explanation behavior of Logistic Regression is clearer at word level.

4.6 Explanation Sample Selection

The explanation analysis is done on a chosen subset of the test set. The final explanation sample consists of 792 test articles. This sample contains correct and incorrect classifications. It is important to include misclassified examples as explanation analysis should not only be done on successful predictions. Misclassified cases can be used to determine if SHAP and LIME still give consistent explanations when the model is wrong.

The explanation sample is as evenly distributed as possible between the two classes. It consists of 400 fake-news samples and 392 real-news samples. There are 634 articles that are correctly classified and 158 articles that are misclassified. This design enables the explanation evaluation to cover both normal and error cases.

Table 5. Explanation sample composition

Category	Count
Total explanation samples	792
Fake-news samples, label 0	400
Real-news samples, label 1	392
Correctly classified samples	634
Misclassified samples	158
Predicted fake, label 0	386
Predicted real, label 1	406

Table 5 indicates that the explanation sample is almost balanced with respect to true label and predicted label**.** It also has a large number of misclassified cases. This strengthens the evaluation of the explanation as consistency is checked in various prediction situations.

4.7 SHAP Configuration

SHAP is used to obtain feature level contribution scores for each selected test sample. As the final model is a linear TF-IDF Logistic Regression classifier, SHAP can give word-level contribution values to the model prediction [10]. SHAP explanation generation is based on a background sample of 200 training records.

One important adjustment is made when interpreting SHAP. SHAP values can naturally describe the direction of class 1 in binary classification. However, the model may predict either class 0 or class 1. SHAP values are aligned with the predicted class to make SHAP comparable with LIME. The SHAP direction remains the same if the predicted class is 1. When the predicted class is 0, the SHAP signs are inverted to make sure that the explanation is for the predicted class. This is to avoid unfair comparison between SHAP and LIME.

4.8 LIME Configuration

LIME is applied to create local explanations at the word level for the same test samples selected [14]. LIME provides a prediction of the class for each article, not a predetermined class. This is important because it is to compare both explanation methods for the same model decision.

The LIME configuration is set to 2,000 perturbation samples per explanation and to return up to 30 important features. LIME is also executed multiple times for stability assessment. Repeated LIME explanations are produced for each article with different random seeds. These repeated explanations are compared to see if LIME outputs the same results for the same input.

Table 6. SHAP and LIME explanation configuration

Setting	SHAP	LIME
Explanation target	Predicted class after alignment	Predicted class
Feature level	Word-level TF-IDF unigram	Word-level text tokens
Number of explanation samples	792	792
Maximum returned features	Top-K selected from contribution scores	30 features before Top-K selection
Main Top-K setting	10	10
Additional Top-K settings	5, 15, 20	5, 15, 20
Background / perturbation setting	200 training samples	2,000 perturbation samples
Repeated-run stability	Not repeated	5 repeated runs

Table 6 shows the SHAP and LIME explanation settings. **The key is that both methods are aligned to the predicted class prior to comparison.** This renders the explanation consistency evaluation more valid.

4.9 Feature Normalization Before Comparison

Before computing consistency metrics, SHAP and LIME features are normalized. This is required as the same word can be used in different forms in both methods of explanation. For instance, one method might return “Reuters”, another “reuters” or “reuters”. If not normalized, these would be considered as different features, but they are actually the same word.

The normalization process includes:

1. Lowercasing feature text.
2. Removing punctuation.
3. Normalizing whitespace.
4. Removing empty features.
5. Normalized duplicate features (addition of contribution scores).

Normalization is done and the features are ranked based on the absolute contribution score. Top-5, Top-10, Top-15 and Top-20 features are chosen for consistency evaluation.

4.10 Evaluation Metrics

Two sets of evaluation metrics are used in the experiment: classification metrics and explanation reliability metrics. Classification metrics assess the predictive performance of the fake news detection model, and explanation reliability metrics assess the consistency, stability and faithfulness of SHAP and LIME explanations.

4.10.1 Classification Metrics

The accuracy, balanced accuracy, precision, recall, F1-score, ROC-AUC, PR-AUC, MCC, and confusion matrix values are used to evaluate the classification model. Suppose TP, TN, FP, FN are the number of true positives, true negatives, false positives, and false negatives, respectively.

Accuracy is defined as the percentage of correct classifications overall:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Precision is the number of positive samples predicted as positive divided by the total number of samples predicted as positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall is the number of actual positive samples that are correctly identified:

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The F1-score is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times \text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}}$$

Specificity is the percentage of actual negative samples that are correctly identified:

$$\text{Specificity} = \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

Balanced accuracy is the mean of recall and specificity:

$$\text{Balanced Accuracy} = \frac{\text{Recall} + \text{Specificity}}{2}$$

Matthew's correlation coefficient is a measure of binary classification that takes into account all four values of the confusion matrix:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

ROC-AUC is the area under the receiver operating characteristic curve**.** The ROC curve is based on the true positive rate and false positive rate:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

PR-AUC is the area under the precision-recall curve, which is generated from precision and recall at various decision thresholds.

4.10.2 Explanation Consistency Metrics

Top-K overlap ratio, Jaccard similarity, Spearman correlation, Kendall correlation and sign agreement are used for the SHAP-LIME comparison.

Let $\boldsymbol{S_K}$ denote the Top-K features set selected by SHAP, and let $\boldsymbol{L_K}$ denote the Top-K features set selected by LIME**.**

Top-K overlap ratio is the ratio of the number of common important features between SHAP and LIME:

$$\mathrm{OR}_K = \frac{|S_K \cap L_K|}{K}$$

Strict set-level similarity between SHAP and LIME feature sets is measured by Jaccard similarity:

$$J(S_K, L_K) = \frac{|S_K \cap L_K|}{|S_K \cup L_K|}$$

Spearman correlation is used to assess the rank-level agreement between SHAP and LIME feature scores:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

where $\boldsymbol{d_i}$ is the difference between the SHAP and LIME ranks of feature $\boldsymbol{i}$, and $\boldsymbol{n}$ is the number of compared features.

The Kendall correlation coefficient is used to measure the agreement between the pairwise rankings of SHAP and LIME:

$$\tau = \frac{C - D}{C + D}$$

where $\boldsymbol{C}$ is the number of concordant feature pairs and $\boldsymbol{D}$ is the number of discordant feature pairs.

Sign agreement: Do SHAP and LIME agree on the direction of contribution of shared features:

$$\mathrm{SA} = \frac{1}{|S_K \cap L_K|} \sum_{i \in S_K \cap L_K} I\left[\mathrm{sign}(S_i) = \mathrm{sign}(L_i)\right]$$

where $\boldsymbol{S_i}$ is the SHAP contribution score for feature $\boldsymbol{i}$, $\boldsymbol{L_i}$ is the LIME contribution score for feature $\boldsymbol{i}$, and I is an indicator function that returns 1 when the condition is true and 0 otherwise.

4.10.3 LIME Stability Metrics

LIME stability is evaluated by repeating LIME explanations several times for every sample. Let $\boldsymbol{L_K^{(a)}}$ and $\boldsymbol{L_K^{(b)}}$ denote the Top-K LIME feature sets from two repeated runs.

The repeated-run Jaccard stability is defined as:

$$J_{\mathrm{LIME}} = \frac{\left|L_K^{(a)} \cap L_K^{(b)}\right|}{\left|L_K^{(a)} \cup L_K^{(b)}\right|}$$

The repeated-run Spearman stability is:

$$\rho_{\text{LIME}} = \rho\left(L^{(a)}, L^{(b)}\right)$$

The repeated run Kendall stability is computed as:

$$\tau_{\text{LIME}} = \tau\left(L^{(a)}, L^{(b)}\right)$$

The sign agreement of the repeated run is:

$$\text{SA}_{\text{LIME}} = \frac{1}{\left|L_K^{(a)} \cap L_K^{(b)}\right|} \sum_{i \in L_K^{(a)} \cap L_K^{(b)}} I\left[\text{sign}\left(L_i^{(a)}\right) = \text{sign}\left(L_i^{(b)}\right)\right]$$

The mean of the pairwise stability values for all repeated LIME runs is the final LIME stability score:

$$\overline{\text{Stability}} = \frac{1}{M} \sum_{p=1}^{M} \text{Stability}_p$$

where $\boldsymbol{M}$ is the number of pairwise repeated-run comparisons**.**

4.10.4 Faithfulness Metric

The deletion test is used to measure faithfulness. In this test, the Top-10 explanation words are removed from the article text, and the model's predicted-class probability is measured again.

Let $\boldsymbol{P_{\text{original}}}$ be the model's predicted-class probability before removing explanation words, and **let** $\boldsymbol{P_{\text{removed}}}$ be the predicted-class probability after removing the Top-K explanation words**.**

The probability drop is:

$$\Delta P = P_{\text{original}} - P_{\text{removed}}$$

The greater the positive probability drop, the more influential the explanation words removed were in the model's prediction. A negative probability drop indicates that the model's confidence was raised by removing the selected words. This can occur in text classification, where deletion might result in unnatural text or words that were weakening the prediction being removed.

4.10.5 Bootstrap Confidence Intervals

Bootstrap confidence intervals are calculated purely for explanation consistency along with the stability metrics. Let $\boldsymbol{x_1, x_2, \ldots, x_n}$ represent the metric values across the samples of n explanation**.** Bootstrap resampling repeatedly does the sampling for these values with replacement and calculation of the mean for each bootstrap sample.

For bootstrap sample $\boldsymbol{b}$, the mean is calculated as:

$$\bar{x}^{(b)} = \frac{1}{n} \sum_{i=1}^{n} x_i^{(b)}$$

The 95% confidence interval is estimated as the 2.5th and 97.5th percentiles of the bootstrap means after BBB bootstrap iterations:

$$\mathrm{CI}_{95\%} = \left[Q_{2.5}\left(\bar{x}^{(1)}, \ldots, \bar{x}^{(B)}\right),\ Q_{97.5}\left(\bar{x}^{(1)}, \ldots, \bar{x}^{(B)}\right)\right]$$

These intervals indicate if the reported explanation metrics are stable over the explanation sample. Small confidence intervals show that the metric estimates are likely to be accurate.

4.11 Implementation Environment

The experiment is implemented in Python using Google Colab. Persistent storage is provided by Google Drive to allow the experiment to be continued after runtime interruptions**.** These are the files that are saved during the implementation: Dataset files, Trained models, Vectorizers, Prediction results, Explanation files, Consistency metrics, Plots, and Final report assets**.** This is important since SHAP/LIME explanation generation can be time consuming for hundreds of text samples.

The main libraries that are used in this experiment include scikit-learn for model training and evaluation, SHAP for Shapley-value-based explanation generation, LIME for local text explanations, NumPy and pandas for data processing, and Matplotlib for visualization**.** The code stores intermediate results after small sets of samples, minimizing the loss of work in case of Colab disconnections.

4.12 Summary

The experimental setup aims to provide a fair and reproducible assessment of SHAP-LIME explanation consistency**.** The data set is well prepared; the classifier is selected based on the validation results and the final model is tested on unseen data. SHAP and LIME explanations are created for the same prediction samples, for the same predicted class, at the feature level and assessed with multiple reliability metrics. This setup provides a strong basis for the results and discussion presented in the next section.

5. Results and Discussion

This section shows the experimental results of the proposed fake news detection and explanation consistency framework. The discussion is broken up into four sections. First, the classification performance of the selected model is analyzed. Second, the confusion matrix, ROC curve and precision-recall curve are discussed. Third, explanation consistency results of SHAP-LIME are shown for varying Top-K values. Lastly, LIME stability, bootstrap confidence intervals, and faithfulness deletion results are discussed.

5.1 Classification Performance of the Selected Model

The final selected model is a balanced Logistic Regression with 80,000 unigram TF-IDF features. The model has a decision threshold set at 0.52, which was optimized during validation. This model was chosen because it had the highest validation F1-score in the model selection process and was appropriate for word-level explanation analysis.

Table 7. Final test performance of the selected model

Metric	Value
Accuracy	96.06%
Balanced accuracy	96.03%
Precision	95.61%
Recall	95.69%
F1-score	95.65%
ROC-AUC	99.31%
PR-AUC	99.18%
MCC	92.05%
Tuned threshold	52%
Test samples	12,710
TF-IDF features	80,000

As can be seen from Table 7, the selected model has a good performance on the unseen test set. The accuracy of 96.06% indicates that the model is able to correctly classify approximately 96% of the test articles. The balanced accuracy of 96.03% is very close to the overall accuracy, which indicates that the model is not biased towards any particular class. The precision value of 95.61% and recall value of 95.69% are also close to each other. This means that the model has a good balance between not predicting positive class samples as negative and not predicting negative class samples as positive. This balance is reflected in the F1-score of 95.65%. The MCC value of 92.05% also reinforces the reliability of the model as it takes into account all four outcomes of the confusion matrix. In binary classification, a high MCC value is particularly valuable as it offers a more balanced performance perspective than accuracy alone.

The ROC-AUC value is 99.31% and PR-AUC value is 99.18%, which indicates that the model has a strong separation between fake and real articles based on probability. The results show that the classifier is sufficiently robust to be used in explanation analysis. As the primary goal of this paper is explanation consistency, a robust classifier is required for the interpretation of SHAP and LIME explanations.

5.2 Confusion Matrix Analysis

The confusion matrix provides a more detailed picture of the errors made by the model. It displays the number of fake and real articles correctly or incorrectly classified.

Table 8. Confusion matrix of the selected model

Actual class	Predicted fake, label 0	Predicted real, label 1
Fake, label 0	6,705	253
Real, label 1	248	5,504

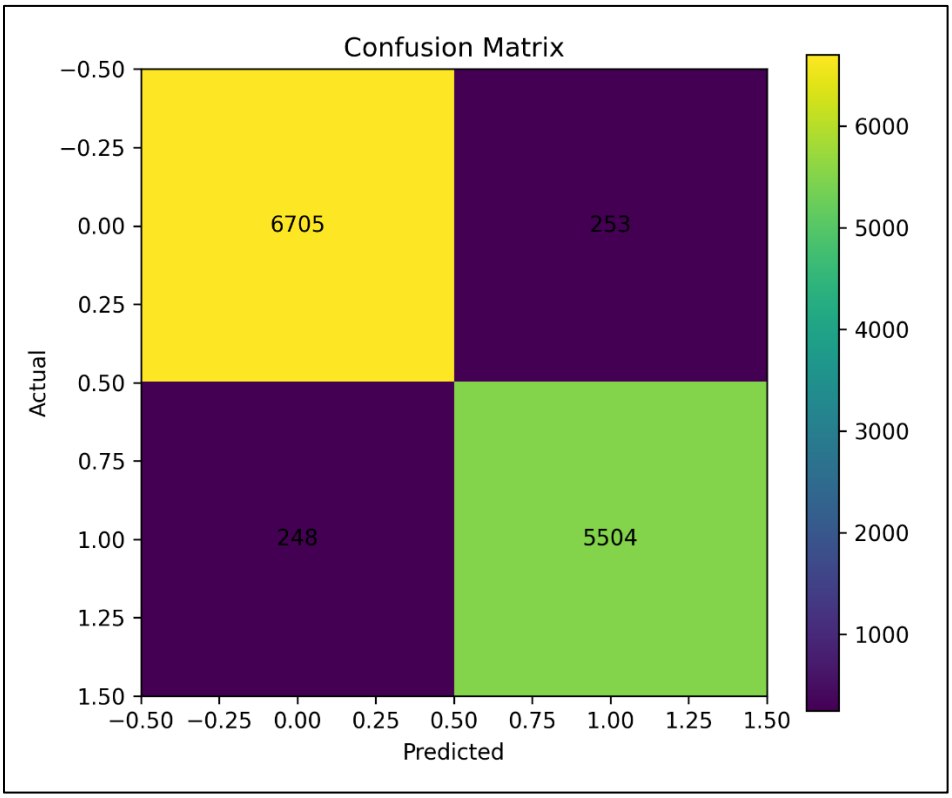


Fig. 4. Confusion matrix of the selected fake news detection model.

The confusion matrix of the proposed fake news detection model is shown in figure 4. The model correctly classifies 6,705 fake articles as fake and 5,504 real articles as real. It incorrectly classifies 253 fake articles as real and 248 real articles as fake. The diagonal values are significantly larger than the off-diagonal values, which is evident from the figure and indicates that the model is performing well for both the classes. The error distribution is also crucial for fake news detection. Fake real errors can be dangerous, as fake articles can be circulated as real news. False-fake errors are also detrimental as they can lead to the rejection of real articles, thereby decreasing trust in real information. The model makes 253 false-real errors and 248 false-fake errors in this experiment. Both these error counts are very close to each other and are much smaller than the number of correctly classified samples, indicating that the classifier is not strongly biased towards one class.

The confusion matrix, overall, confirms the balanced performance indicated in the evaluation results. The model is able to correctly classify fake articles with a recall of about 96.36% and real articles with a recall of about 95.69%. Overall accuracy is ~96.06% and the balanced accuracy is ~96.03%. Both classes are predicted with similar reliability, so the model can be used for the next step of SHAP and LIME explanation consistency evaluation.

5.3 ROC and Precision-Recall Curve Analysis

The ROC curve and precision-recall curve are used to evaluate the selected model based on probability.

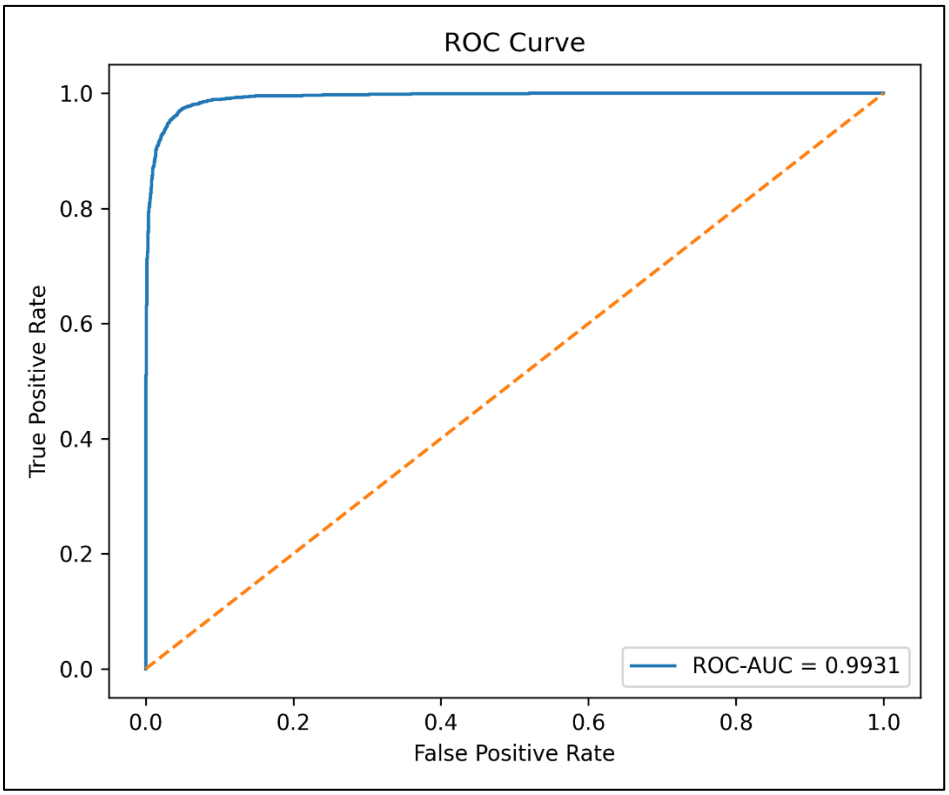


Fig. 5. ROC curve of the selected fake news detection model.

The ROC curve of the proposed fake news detection model is shown in Figure 5. The blue curve is still near the top left corner of the plot and the orange dashed diagonal line is the random classification performance. The ROC curve is well above the random line, indicating that the model has a high discriminatory power between fake and real articles at various decision thresholds. The ROC-AUC value of the selected model is 0.9931 (or 99.31%). This is a very high score, which means that the model is very effective in separating the two classes. A ROC-AUC value near 1.0 indicates that the model is able to predict the correct class with a higher score than the incorrect class in most cases. Hence, the model has a good discrimination power for fake news detection.

Overall, the results shown in Figure 5 corroborate the classification results reported in the evaluation table. The high ROC-AUC value indicates that the model is not only accurate at one fixed threshold but also is reliable at varying thresholds. This allows the classifier to be further explained using SHAP and LIME in terms of consistency.

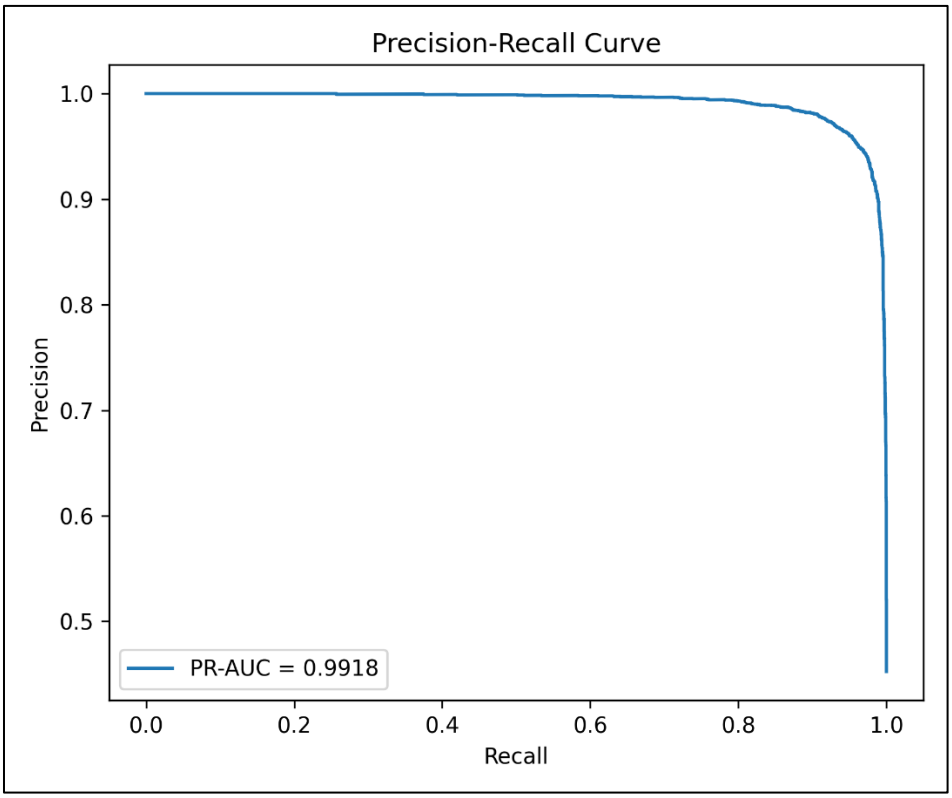


Fig. 6. Precision-recall curve of the selected fake news detection model.

The precision-recall curve of the proposed fake news detection model is shown in figure 6. For the majority of recall values, the curve is close to the top of the graph, indicating that the model has a high precision even at high recall values. This implies that the classifier is able to classify a lot of relevant fake or real articles with a low number of incorrect positive predictions.

The PR-AUC value for the selected model is 0.9918 (or 99.18%). This is a very good result and validates the good performance of the model for various classification thresholds. PR-AUC is particularly helpful in the context of fake news detection as datasets may not always have perfectly balanced class distributions. A high PR-AUC value indicates that the model is able to maintain a good balance between precision and recall rather than just doing well due to the majority class.

The combination of ROC curve and precision-recall curve indicates that the model chosen has good classification ability. The ROC-AUC shows good class separation and the PR-AUC shows good precision-recall behaviour. These results indicate that the trained model can be used in the next phase of the study to generate SHAP and LIME explanations and compare the explanations for consistency.

5.4 Explanation Sample Analysis

The explanation analysis is carried out on 792 test samples selected. The sample contains correctly classified and misclassified articles, making the evaluation of the explanation more realistic. The explanation results would not illustrate the behavior of SHAP and LIME when the model is wrong if only correctly classified samples were used.

Table 9. Explanation sample composition

Category	Count
Total explanation samples	792
Fake-news samples, label 0	400
Real-news samples, label 1	392
Correctly classified samples	634
Misclassified samples	158
Predicted fake, label 0	386
Predicted real, label 1	406

Table 9 indicates that the explanation sample is almost evenly split between fake and real articles. It has 400 fake-news samples and 392 real-news samples. It also has 634 samples correctly classified and 158 samples misclassified. This is helpful as it enables the evaluation of the consistency of explanation for both successful and unsuccessful predictions. The distribution of the predicted labels is also balanced, with 386 samples being predicted as fake and 406 samples being predicted as real. This completely avoids the explanation comparison that is being dominated by one predicted class.

5.5 SHAP-LIME Consistency Across Top-K Values

This study aims to assess the consistency of the explanations given by SHAP and LIME for the same fake news predictions. The comparison is done after predicted-class alignment and feature normalization**.** This correction is crucial as SHAP and LIME might otherwise describe different directions of classes or different forms of features.

Table 10. SHAP-LIME consistency summary across Top-K values

Top-K	Overlap ratio (%)	Jaccard similarity (%)	Spearman correlation (%)	Kendall correlation (%)	Sign agreement (%)
5	77.35%	65.78%	74.60%	66.56%	98.05%
10	76.94%	64.35%	80.42%	70.32%	98.25%
15	75.18%	62.03%	81.31%	70.33%	98.35%
20	74.68%	61.56%	81.17%	69.57%	98.49%

After correction, SHAP and LIME have strong agreement as shown in Table 10. At Top-10, the overlap ratio is 76.94%. This implies that approximately 77% of the Top-10 explanation words are common to both SHAP and LIME. This is a good result as the two methods have different explanation mechanisms. SHAP is based on Shapley-value contribution scores [10], while LIME is based on local perturbation and local surrogate modeling [14]. The Jaccard similarity at Top-10 is 64.35%. This value is less than the overlap ratio as Jaccard similarity is stricter. It is the ratio of the common features to the union of the two sets of features. Hence, if both the methods contain several important words, but also some unique words, then the Jaccard similarity may be low. The Spearman correlation at Top-10 is 80.42%, and the Kendall correlation is 70.32%. There is good agreement between the rankings of these values. This implies that SHAP and LIME not only choose many of the same words, but also prioritise important words in a similar manner. The sign agreement at Top-10 is very high

(98.25%). This indicates that SHAP and LIME choose the same word, they tend to agree on whether that word supports or opposes the predicted class almost 100% of the time.

The results are consistent from Top-5 to Top-20. The overlap ratio slightly decreases from 77.35% at Top-5 to 74.68% at Top-20. This is natural as more dominant features are excluded in the increase of Top-K and explanation methods can vary more. But the cut is modest and the pact is robust. Directional agreement is also very stable, with a slight increase from 98.05% to 98.49% when signing agreement.

5.6 Visual Analysis of Top-10 Consistency

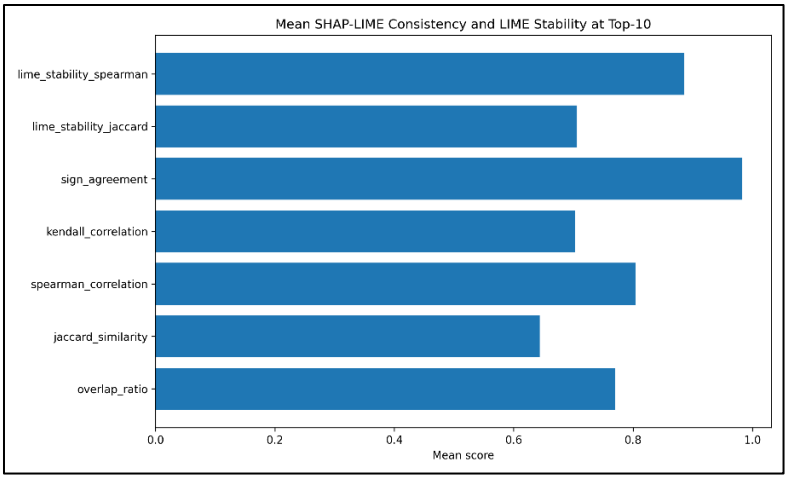


Fig. 7. Mean SHAP-LIME consistency and LIME stability metrics at Top-10.

The main Top-10 explanation consistency and stability metrics are visually compared in figure 7. As can be seen from the figure, the best overall result is signing agreement. The SHAP-LIME sign agreement is 0.9825, or 98.25%, while the LIME repeated-run sign agreement is 0.9987, or 99.87%. The very high values indicate that the explanation methods almost always agree when they find common important words, whether they support or oppose the predicted class. There is also a high level of agreement in the figure's ranking. The SHAP-LIME Spearman correlation is 0.8042 or 80.42% which shows a strong rank level agreement between the two explanation methods. This indicates that SHAP and LIME not only select many of the same important words, but they also order them in a fairly similar fashion. The Kendall correlation is 0.7032 (or 70.32%) which further indicates this ranking consistency.

The Jaccard similarity is lower (0.6435 or 64.35%) because it is based on exact same set-level agreement and hence it is usually stricter than rank-based comparison. This is a significant trend. Although the Top-10 feature sets chosen by SHAP and LIME are not necessarily the same in all cases, there is strong agreement on the direction of the features when they share features, and the rankings of the features are closely related.

The LIME stability results also strengthen the overall reliability of the explanation process. The LIME stability Spearman is about 88.52% and LIME stability Jaccard is about 70.62%, indicating that repeated LIME explanations are reasonably stable across runs. Figure 6 shows that, overall, SHAP and LIME are reasonably consistent in the explanations they provide for the selected fake news detection model after fair predicted-class alignment and feature.

5.7 Consistency Trends Across Top-K Values

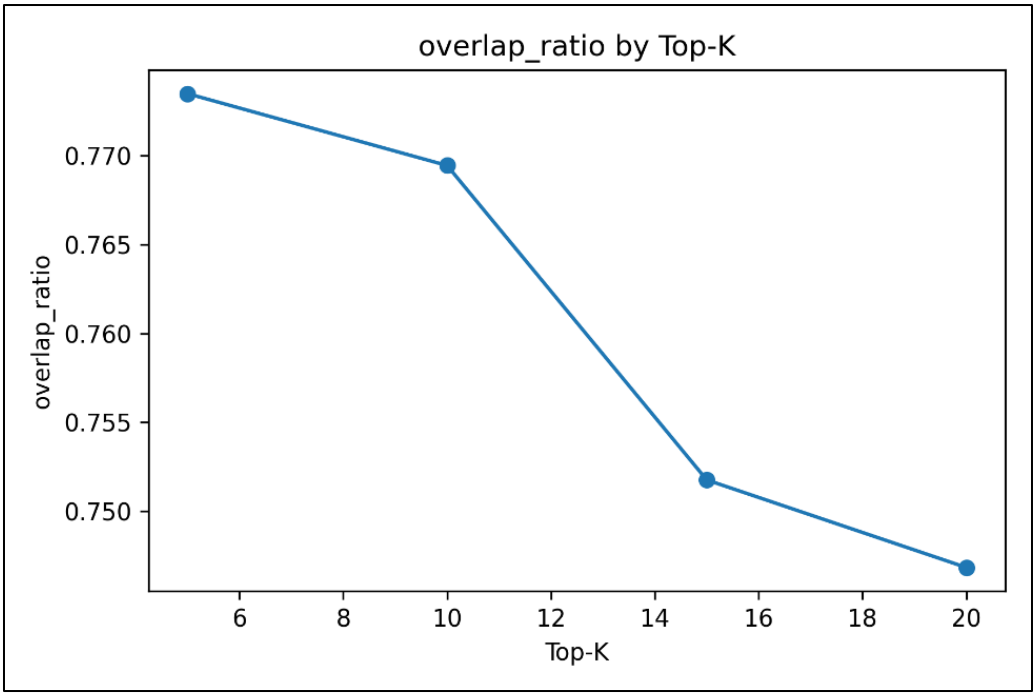


Fig. 8. SHAP-LIME overlap ratio across Top-K values.

The overlap ratio between SHAP and LIME explanations for various Top-K values is shown in Figure 8. The overlap ratio remains high for all tested values, with 0.7735 at Top-5, 0.7694 at Top-10, 0.7518 at Top-15, and 0.7468 at Top-20. This translates to a percentage of around 77.35%, 76.94%, 75.18%, and 74.68% of the selected explanation features at these Top-K levels, respectively, for SHAP and LIME. The curve is slightly downward sloping with an increase in Top-K.

This implies that SHAP and LIME are most aligned on the most important and most influential words, which are the top-ranked ones. The agreement decreases slightly when more lower ranked features are added, as these features are typically less dominant and could be chosen differently by each of the explanation methods. But the drop is minimal and the overlap ratio is still greater than 74% at Top-20. This means that there is a high level of agreement between the two explanation methods at the feature level at different explanation depths. Thus, it can be said that the explanations generated by SHAP and LIME are reasonably consistent for the selected fake news detection model as shown in Figure 7.

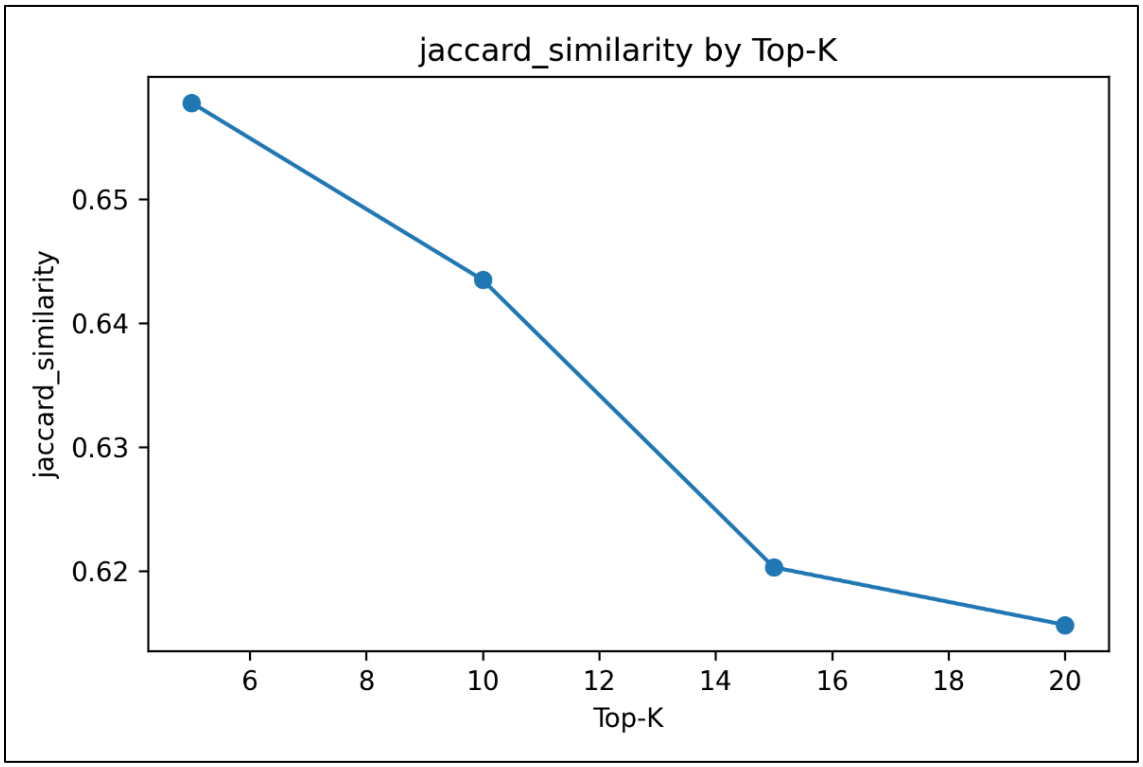


Fig. 9. SHAP-LIME Jaccard similarity across Top-K values.

The Jaccard similarity between SHAP and LIME explanations for varying Top-K values is shown in Figure 9. The Jaccard similarity decreases from 0.6578 at Top-5 to 0.6156 at Top-20. This translates to a percentage drop in the feature-set agreement from about 65.78% to 61.56% as the number of features increases. This downward trend is to be expected as Jaccard similarity is a strict set-level measure. It compares the common features to the union of all the features selected by SHAP and LIME. The larger the Top-K, the more features will be selected. As such, even slight variations in the vocabulary used by each method can lower the Jaccard score.

The values are still above 60% for all Top-K settings, though, which still shows a meaningful agreement in the feature-sets. This implies that sometimes SHAP and LIME do not choose exactly the same explanation words, but they still share a significant number of important features. Thus, Figure 8 confirms that SHAP and LIME explanations are fairly consistent across the different levels of explanation.

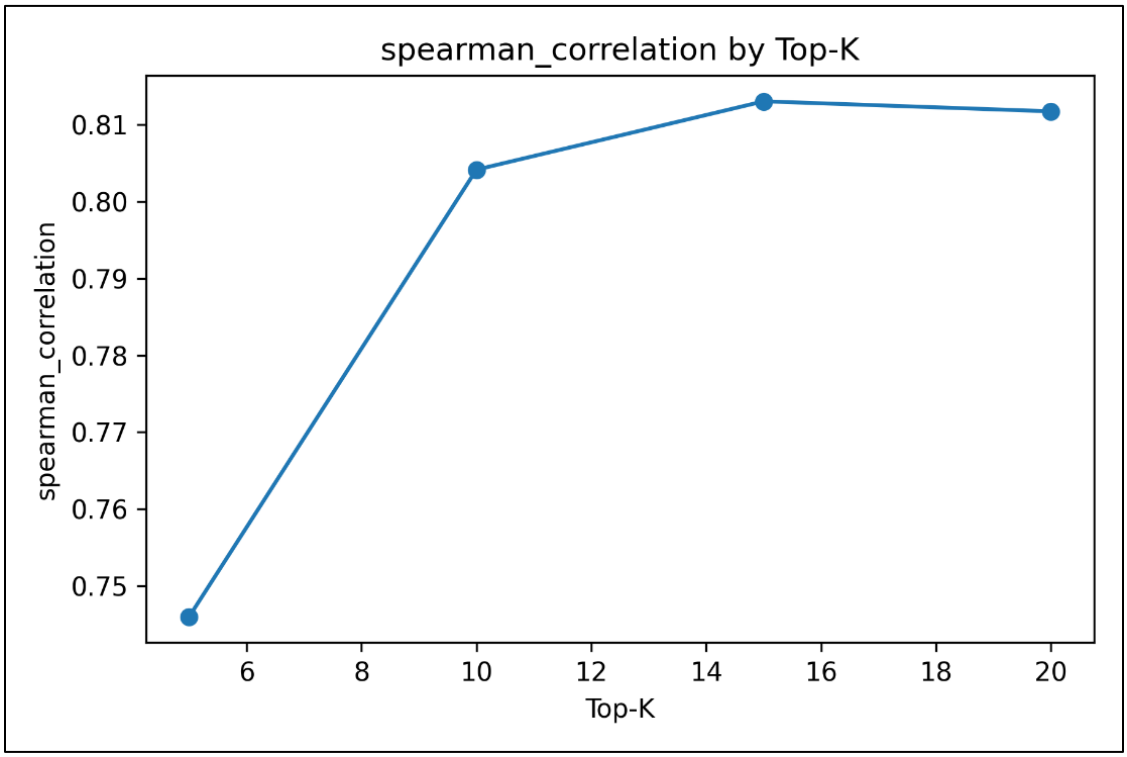


Fig. 10. SHAP-LIME Spearman correlation across Top-K values.

The Spearman correlation between SHAP and LIME explanations for various Top-K values is shown in Figure 10. The Spearman correlation increases from 0.7460 at Top-5 to 0.8042 at Top-10. It then maintains the state to be above 0.81 at Top-15 and Top-20. This suggests that it corresponds to a rank-level agreement of approximately 74.60% at Top-5, 80.42% at Top-10, and more than approximately 81% at larger Top-K. This trend indicates that the more explanation features that are included, the more rank-level agreement that is achieved. The five most important words may not be in the same order for SHAP and LIME as the top features are sensitive to the scoring mechanism used by each method. But if more explanation words are taken into account, the overall ranking pattern is more similar.

The overall results of Figure 10 indicate that SHAP and LIME have high ranking consistency across explanation depths. The Spearman correlation results show that the importance rankings of the two methods are reasonably similar for the selected fake news detection model, as the correlation increases and stabilizes with the increase in number of features.

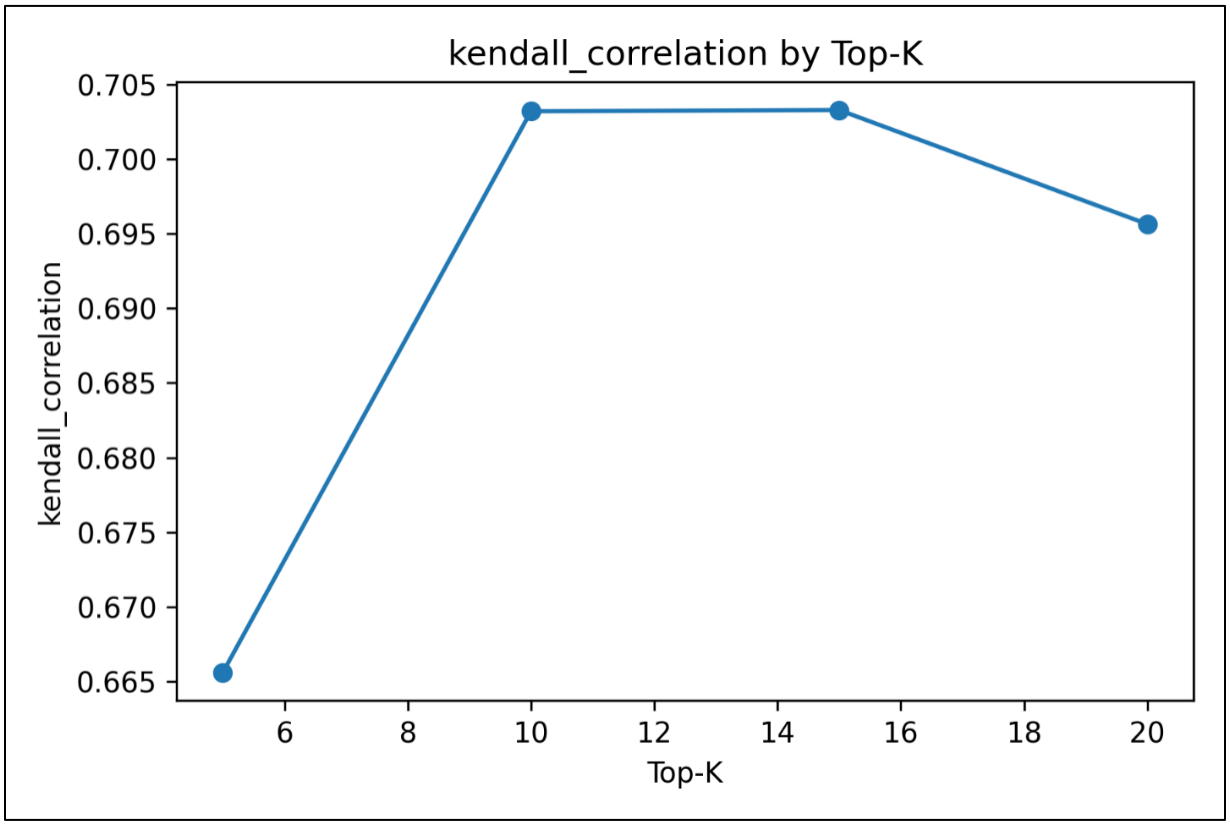


Fig. 11. SHAP-LIME Kendall correlation across Top-K values.

The Kendall correlation between SHAP and LIME explanations for different Top-K values is shown in Figure 11. The Kendall correlation is interesting as it rises from around 0.655 at Top-5 to around 0.703 at Top-10 and Top-15, and then it stays around 0.696 at Top-20**.** This corresponds to a percentage of approximately 65.5% at Top-5, 70.3% at Top-10 and Top-15, and 69.6% at Top-20. This trend indicates that SHAP and LIME have consistent ordering agreement with the addition of more explanation features. Kendall correlation is stricter than Spearman correlation since it measures the pairwise ordering agreement between ranked features. Thus, values in the range of 0.70 suggest strong ordinal consistency between the two methods, and not just a general relationship between the two methods in terms of their ranking.

In general, Figure 11 confirms the conclusion that SHAP and LIME explanations are fairly consistent for different Top-K values. The Top-5 Kendall value is slightly lower but the values for Top-10 to Top-20 are stable around 0.70 indicating that both methods exhibit similar feature-ordering behaviour at wider explanation depths.

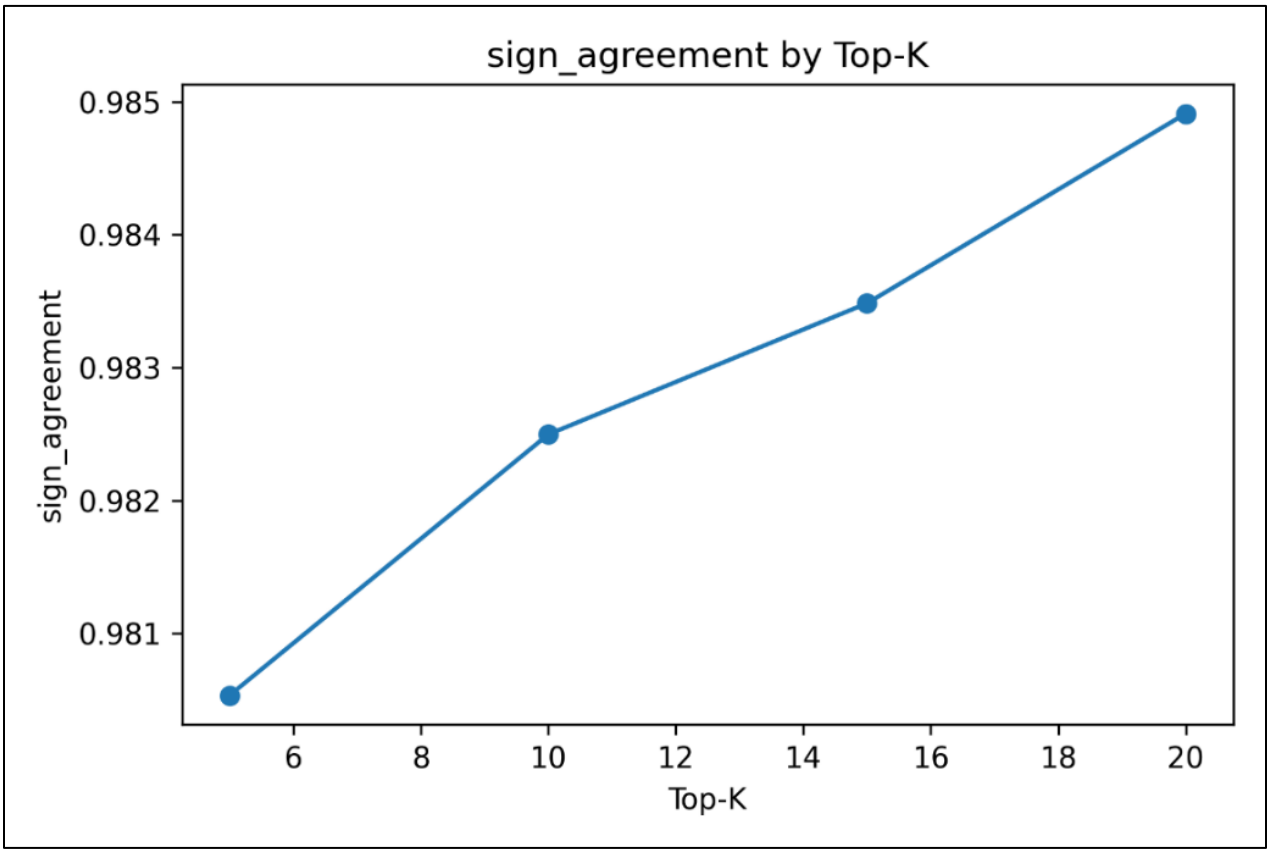


Fig. 12. SHAP-LIME sign agreement across Top-K values.

The sign agreement between SHAP and LIME explanations for various Top-K values is shown in Figure 12. The sign agreement is always very high, ranging from 0.9805 at Top-5 to 0.9849 at Top-20. This translates to an increase in directional agreement from ~98.05% to 98.49% when more explanation features are taken into account. This finding implies that SHAP and LIME select the same word as important almost always agree on the direction of the word's effect. That is, both methods typically agree on whether the shared word is in favor of the predicted class or not. This is significant because explanation consistency is not only the choice of similar words, but also the choice of similar meaning for the words.

In general, Figure 12 is one of the most positive results of the study. The sign agreement is very high and stable, indicating that SHAP and LIME generate very consistent explanations in the direction of the Top-K setting. This firmly validates the fact that both methods give reliable explanations for the selected fake news detection model, without contradicting each other, after predicted-class alignment and feature normalization.

5.8 LIME Repeated-Run Stability

Repeated-run stability is important as LIME can be sensitive to random perturbation sampling [14]. This experiment is repeated five times for each explanation sample with different random seeds for LIME.

Jaccard similarity, Spearman correlation, Kendall correlation and sign agreement are used to compare the repeated explanations.

Table 11. LIME repeated-run stability summary

Stability metric	Mean value (%)
LIME stability Jaccard	70.62%
LIME stability Spearman	88.52%
LIME stability Kendall	79.57%
LIME stability sign agreement	99.87%

Table 11 indicates that LIME explanations are reasonably stable when repeated runs are performed. The Jaccard stability score of 70.62% indicates that repeated LIME explanations tend to choose similar important words. The Spearman stability value is 88.52% indicating high rank level stability and the Kendall value is 79.57% indicating high ordinal agreement. The sign agreement value is very high (99.87%) which indicates that repeated LIME runs that select the same feature almost always assign the same contribution direction. These results show that LIME is stable under the selected configuration. This is crucial as unstable explanations would reduce the reliability of the SHAP-LIME comparison. LIME provides the same explanations on multiple runs, making the consistency analysis more reliable.

5.9 Bootstrap Confidence Interval Analysis

To assess the statistical reliability of the explanation metrics, bootstrap confidence intervals are computed. The Top-10 confidence intervals are shown in Table 12.

Table 12. Bootstrap confidence intervals for Top-10 explanation consistency

Metric	Mean	95% CI lower	95% CI upper	Samples
Overlap count	7.6944	7.6061	7.7841	792
Overlap ratio	76.94%	76.10%	77.80%	792
Jaccard similarity	64.35%	63.18%	65.46%	792
Spearman correlation	80.42%	79.11%	81.60%	792
Kendall correlation	70.32%	69.04%	71.49%	792
Sign agreement	98.25%	97.81%	98.62%	792
LIME stability Jaccard	70.62%	69.61%	71.67%	792
LIME stability Spearman	88.52%	87.64%	89.42%	792
LIME stability Kendall	79.57%	78.56%	80.57%	792
LIME stability sign agreement	99.87%	99.66%	99.99%	792

Table 12 shows narrow confidence intervals for all main Top-10 metrics. For instance, the 95% confidence interval for overlap ratio is [76.10%, 77.80%], indicating that the estimated overlap of the features is stable in the explanation sample. The rank agreement is consistently strong with the confidence interval for Spearman correlation being 79.11% to 81.60%. The narrow and very high confidence interval for sign agreement is 97.81% to 98.62%. These confidence intervals provide added confidence in the results. They demonstrate that the high SHAP-LIME consistency is not due to a small number of samples but is consistent across the 792 explanation cases.

5.10 Faithfulness Deletion Test

The faithfulness deletion test tests if the explanation words have an impact on the model's prediction. In this test, the Top-10 words selected by SHAP and LIME are removed from the text and the probability of the model's predicted class is measured again. A lower probability indicates that the words that were removed were significant to the model's confidence.

Table 13. Faithfulness deletion test using Top-10 explanation words

Method	Mean probability drop (%)	Median probability drop (%)	Standard deviation (%)	Minimum (%)	Maximum (%)
SHAP	13.94%	8.49%	19.89%	-41.31%	90.11%
LIME	12.93%	7.89%	17.61%	-32.45%	79.85%

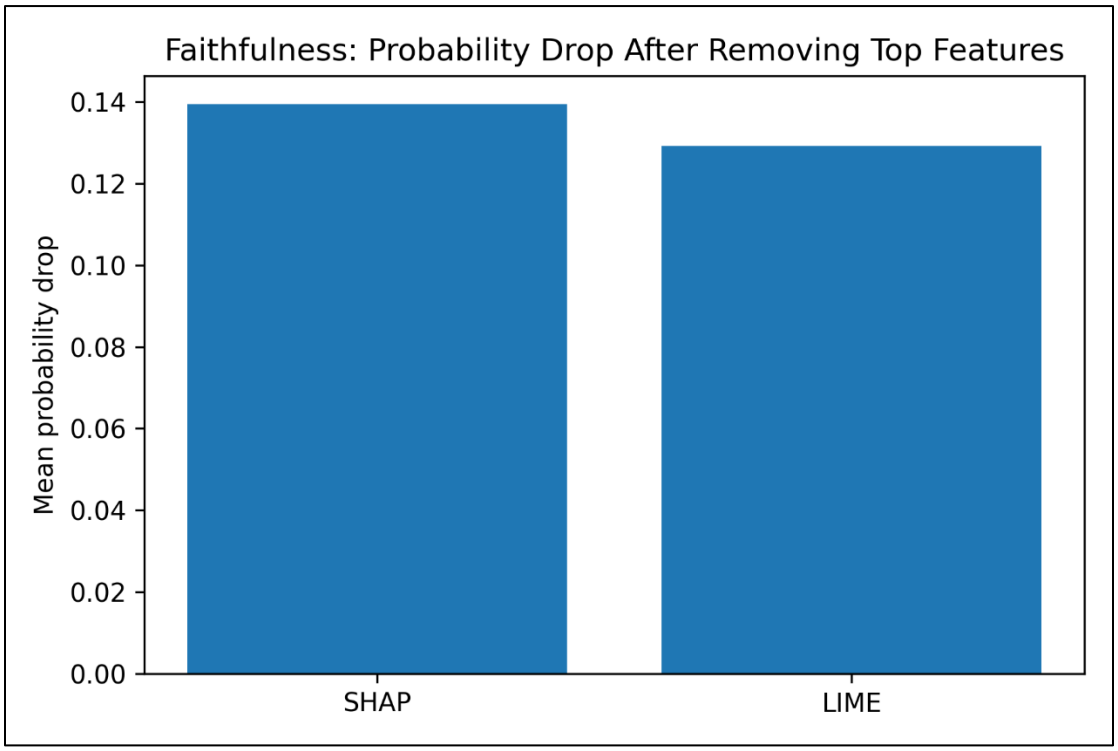


Fig. 13. Faithfulness deletion test based on Top-10 explanation words.

As can be seen in Table 13 and Fig. 13, the probability of the predicted class for both SHAP and LIME decreases when the Top-10 explanation words are removed. The average probability decrease is 13.94% for SHAP and 12.93% for LIME. This indicates that, on average, SHAP-selected words lead to a slight decrease in model confidence when removed. The median drops are 8.49% for SHAP and 7.89% for LIME, indicating that the effect is not limited to a few extreme cases. The minimum values for both methods are negative, however. That is, sometimes the removal of the selected explanation words actually leads to an increase in the model's confidence. This does not imply that the explanations are incorrect. For text classification, removing words may result in unnatural input text, alter the sentence structure, or remove words that were also hurting the prediction. Thus, the deletion test should be viewed as evidence and not as an absolute measure of faithfulness of explanation. The faithfulness results corroborate the

explanation quality of SHAP and LIME overall. The highlighted words are those that have a measurable impact on the model's confidence and SHAP demonstrates a marginally greater average probability drop than LIME.

5.11 Overall Discussion

The experimental results indicate that the chosen fake news detection model has a strong performance on the WELFake test set. The accuracy of the classifier is 96.06% and the F1 score is 95.65% with high values of ROC-AUC and PR-AUC. The confusion matrix indicates that the error behavior is well balanced with 253 false-real and 248 false-fake predictions. More importantly, the corrected explanation comparison demonstrates good agreement between SHAP and LIME. The two methods have approximately 77% overlap of important words at Top-10 and the agreement in ranking is strong. The sign agreement for the very high case is very high, indicating that the two methods tend to agree on the direction of interpretation for the shared words. This finding is important because SHAP and LIME are based on different explanation principles [10], [14]. For this TF-IDF Logistic Regression model, the agreement indicates that in many cases both methods select the same textual evidence.

The results also suggests that the correction steps are significantly important. If predicted-class alignment and feature normalization is not applied, the agreement may be underestimated when comparing SHAP and LIME. They also need evidence that is comprehensible and credible to support the prediction**.** The proposed implementation does not suffer from this issue since it is aligned to the predicted class, and word-level features are normalized prior to comparison. This renders the explanation evaluation more just and dependable. The results of LIME stability further validate the explanation analysis. LIME explanations are repeatable, meaning that the SHAP-LIME comparison is less susceptible to random variation. The explanation results are also corroborated by the faithfulness deletion test, which shows that on average, removing important words decreases model confidence.

To sum up, the results demonstrate the effectiveness of the proposed framework for classification performance and explanation reliability. When carefully and fairly compared, the study shows that SHAP and LIME can generate consistent explanations for fake news detection.

6. Conclusion and Future Work

This study provided a consistency-based evaluation of SHAP and LIME explanations for fake news detection using machine learning. The primary aim was not just to create a robust fake news classifier, but also to investigate whether two widely used explanation techniques offer consistent and reliable explanations for the same model predictions. This is crucial as fake news detection is a sensitive task and the user requires more than just a predicted label. They also require evidence that is fully trustworthy and understandable in support of the stated prediction.

The WELFake dataset, which includes fake and real news samples at the article level [15] [16] was used for the experiment. The final cleaned data set comprised of 63,547 articles. Stratified splitting was used to split the dataset into training, validation and test sets. The machine learning pipeline used was TF-IDF based and multiple models were tested. The final model chosen was a balanced Logistic Regression classifier with 80,000 unigram TF-IDF features and a decision threshold of 0.52 that was optimized on the validation set. The chosen model had a good classification accuracy on the test set. It obtained 96.06%

accuracy, 95.65% F1-score, 99.31% ROC-AUC, 99.18% PR-AUC, and 92.05% MCC. The confusion matrix also revealed a balanced distribution of errors, with 253 fake articles being classified as real and 248 real articles being classified as fake. From these results, it can be concluded that the trained classifier is reliable enough for further explanation analysis. The main contribution of the study is the corrected SHAP-LIME explanation comparison. The implementation aligns both explanation methods to the same predicted class before comparison. This is important because, in binary classification, SHAP values may naturally describe the direction of class `1`, while LIME explains the predicted class. Without this correction, the comparison can become unfair and misleading. The study also normalizes explanation features by lowercasing, removing punctuation, normalizing spaces, and merging duplicate feature forms. This makes SHAP and LIME explanations more directly comparable at the word level.

The corrected SHAP-LIME explanation comparison is the main contribution of the study. The implementation matches both explanation approaches to the same anticipated class prior to the comparison. This is crucial since in binary classification, SHAP values can naturally describe the direction of class 1, while LIME explains the predicted class. If this correction is not made, the comparison can be completely unfair and misguiding. The study also performs normalization of explanation features, which includes lowercasing, removing punctuation, normalizing spaces and merging duplicate forms of features. This means that the LIME explanations were fairly consistent for the selected setting. SHAP and LIME exhibited high agreement in the corrected explanation results. The overlap ratio was 76.94%, Jaccard similarity was 64.35%, Spearman correlation was 80.42%, Kendall correlation was 70.32% and sign agreement was 98.25% at Top-10. These results show that SHAP and LIME tend to select similar important words and generally agree on whether they are important for the predicted class or not. LIME repeated-run stability was also high, with a Jaccard stability of 70.62% and a Spearman stability of 88.52%. This suggests that the LIME explanations were considerably stable for the chosen configuration.

The explanation results were corroborated by the faithfulness deletion test. The average reduction in the model's predicted-class probability when removing the Top-10 words selected by SHAP and LIME were 13.94% and 12.93%, respectively. This indicates that the highlighted words had a significant impact on the model's confidence. Some samples, however, exhibited negative probability decreases, which indicates that in some cases, the model was more confident when explanation words were removed. This validates that deletion-based faithfulness tests for text classification should be interpreted with caution.

Overall, the results indicate that SHAP and LIME can offer consistent explanations for the fake news detection when carefully compared**.** The results also demonstrate an important evidence that explanation comparison needs proper methodological corrections**.** To make a fair comparison with SHAP-LIME, predicted-class alignment and feature normalization are necessary. Explanation agreement might be underestimated if these steps are not taken.

6.1 Limitations

There are some limitations to this study. First, the experiment is based on one main dataset, WELFake. WELFake can be used for fake news detection at the article level, but may not work on other fake news datasets like LIAR or FakeNewsNet [17, 20**].** The writing style, political topics, source patterns, and other label structures may significantly vary across various datasets. Hence, the same consistency framework should be tested on other datasets in the future. Second, the last classifier is TF-IDF and Logistic Regression. This option is helpful for transparent word-level explanation analysis, but it is not completely

aligned with the current fake news detection systems based on transformer. The explanation behavior of deep learning and transformer models may differ due to the complexity of their internal representations [7], [12], [18]. In the future, SHAP-LIME consistency should be tested for transformer-based models like BERT, RoBERTa, or domain-specific language models. Third, the study adopts unigram TF-IDF features to make the comparison between SHAP and LIME fair. This enhances comparability at the word level, and can miss out on phrase level meaning. Important phrases can be included in fake news, not just individual words. Future research can investigate how to make word-level and phrase-level explanations more comparable, particularly in the case of bigrams, trigrams, and contextual embeddings. Fourthly, the faithfulness deletion test is limited. When you delete words from an article, you may end up with unnatural text and the sentence structure may alter in ways you didn't expect. This can sometimes increase the model's confidence instead of decreasing it. Thus, the deletion test can be employed as a supporting test, but should not be regarded as an absolute indicator of the quality of the explanation. Fifth, this study assesses the consistency of SHAP and LIME, but not the human trust itself. A method can have a very good numerical consistency but yet be hard for end users to understand. Future research needs to involve human evaluation to assess the usefulness and trustworthiness of the explanations for journalists, students, researchers, and general users.

6.2 Future Work

There are a number of avenues for future research that can build on this study. First, the proposed framework can be tested on other fake news datasets like LIAR and FakeNewsNet [17, 20]. This would indicate if the SHAP-LIME consistency is stable for various types of datasets, such as short political statements, article-level news and fake news data based on social context. Second, future research can be done to compare the traditional machine learning models with deep learning and transformer-based models. This would help to identify if there is any change in the explanation consistency as the classifier becomes more complex. The explanations of transformer models may vary from TF-IDF based explanations as transformer models can capture deeper context. Third, in the future, more explanation methods can be added to the research. In addition to SHAP and LIME, other methods like Integrated Gradients, attention-based explanations, Anchors, and counterfactual explanations can be added. This would enable a wider comparison of the reliability of explanation in fake news detection. Fourth, more faithfulness and robustness metrics can be used to expand explanation evaluation. Future work, for instance, can examine faithfulness to insertion, adversarial word replacement, explanation robustness to paraphrasing, and sensitivity to small text changes. These tests would give a better understanding of the reliability of the explanation methods under realistic text variations. Fifth, there may be need to add human-centered evaluations. Users can be asked to evaluate the explanations for understandability, usefulness, and convincingness. This would bridge the quantitative consistency metrics to a real user trustworthiness. As fake news detection is a human use, human evaluation is crucial for practical deployment. Last, future systems can have interactive explanation interfaces. These systems could display many attributes such as the label, confidence score, SHAP and LIME explanation, and consistency score simultaneously. This would help users not only to understand what the model predicted, but also how reliable the explanation seems to be.

6.3 Final Statement

This study shows that the reliability of explanation can be measured systematically in the field of fake news detection. Results show that both SHAP and LIME can produce very consistent explanations when

they are aligned to the same predicted class and compared using normalized word-level features. Hence, the proposed framework can serve as a good starting point to develop fake news detection systems which are not only accurate but also explainable, consistent, and more trustworthy.

REFERENCES

[1] Athira, A. B., et al. (2023). A systematic survey on explainable AI applied to fake news detection. *Engineering Applications of Artificial Intelligence*.

[2] D'Ulizia, A., et al. (2021). Fake news detection: A survey of evaluation datasets. *PeerJ Computer Science*.

[3] Galli, A., et al. (2022). A comprehensive benchmark for fake news detection. *Complex & Intelligent Systems*.

[4] Givisis, I., et al. (2025). Comparing explainable AI models: SHAP, LIME, and their applications. *Electronics*.

[5] Gongane, V. U., et al. (2024). A survey of explainable AI techniques for detection of fake news and hate speech on social media platforms. *Journal of Computational Social Science*.

[6] Hermosilla, P., et al. (2025). Explainable AI for forensic analysis: A comparative study of SHAP and LIME. *Applied Sciences*.

[7] Jadhav, R., et al. (2025). Explainable multilingual and multimodal fake-news detection. *Frontiers in Artificial Intelligence*.

[8] Kozik, R., et al. (2024). Using XAI to fool a fake news detection method. *Computers & Security*.

[9] Kukkar, A., & Kaur, G. (2025). AEC: A novel adaptive ensemble classifier with LIME and SHAP-based interpretability for fake news detection. *Expert Systems with Applications, 281*, 127751.

[10] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*.

[11] Mouratidis, D., et al. (2025). Machine learning strategies for fake news detection. *Information*.

[12] Padalko, H., et al. (2024). A novel approach to fake news classification using LSTM-based deep learning models. *Scientific Reports / related open-access source*.

[13] Rathod, H., Shelar, D., Singh, R., & Modi, N. (2025). Building an explainable and scalable AI system for fake news detection across digital platforms. *International Journal of Computer Applications, 187*(15), 34–42.

[14] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[15] Shahane, S. (2021). *WELFake dataset for fake news detection in text data* [Data set]. Zenodo.

[16] Shahane, S. (n.d.). *Fake news classification / WELFake dataset* [Data set]. Kaggle.

[17] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (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*.

[18] Tian, Y., et al. (2025). An empirical comparison of machine learning and deep learning methods for fake news detection. *Mathematics*.

[19] Vimbi, V., et al. (2024). Interpreting artificial intelligence models: A systematic review of LIME and SHAP. *Journal of Big Data*.

[20] Wang, W. Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

[21] Alshuwaier, F. A., et al. (2025). Fake news detection using machine learning and deep learning: A review. *Computers*.

[22] Trust-Oriented Explainable AI for Fake News Detection. (2026). *arXiv preprint*.