Tabriz Journal of Electrical Engineering (TJEE), vol. 56, no. 1, 2026 Serial no. 115
DOI: 10.22034/tjee.2026.68653.5067

# Improving Fake News Detection Using Machine Learning Methods

Fatemeh Shahidi, Javad Hamidzadeh *

Faculty of Computer Engineering and Information Technology, Sadjad University, Mashhad, Iran
f.shahidy124@sadjad.ac.ir, j_hamidzadeh@sadjad.ac.ir.

*Corresponding author

Received: 15/08/2025, Revised: 06/02/2026, Accepted: 11/04/2026.

## Abstract

The rapid spread of fake news on digital platforms has become a major societal challenge, influencing public opinion, political stability, and social trust. Despite extensive research, existing detection models often suffer from limited generalization, weak robustness to noisy or short texts, and reliance on manually engineered features. Moreover, the evolving linguistic and contextual nature of fake news reduces the long-term effectiveness of conventional approaches. To address these limitations, this study proposes a novel hybrid fake news detection framework that, for the first time, integrates Kernel Fuzzy Rough Set (KFRS)–based feature selection with a two-stage ensemble of deep learning and traditional machine learning classifiers. Unlike prior studies that rely solely on either neural or statistical models, the proposed approach combines LSTM and Bi-LSTM networks with Logistic Regression, Support Vector Machine, and XGBoost, leveraging their complementary strengths through soft voting and stacking. The KFRS component plays a critical role in refining feature representations by handling uncertainty and reducing noise, which is particularly beneficial for short and noisy textual data. Skip-Gram word embeddings are employed to capture semantic relationships between words and enhance contextual understanding. The proposed framework is evaluated on three benchmark datasets—LIAR, FakeNewsNet, and FakeEdited—demonstrating consistently strong performance across diverse domains. Ablation experiments further confirm that incorporating KFRS leads to substantial improvements in Recall and F1-score. Overall, the proposed method offers a robust, scalable, and practically deployable solution for real-world fake news detection.

**Keywords**
Fake news detection, deep learning, machine learning, Kernel Fuzzy Rough Sets, ensemble learning.

## 1. Introduction

The rapid growth of social media, driven by widespread internet and smartphone usage, has transformed how information is disseminated [1]. These platforms now serve as primary, fast, and cost-effective channels for news exchange [2], shaping public discourse and influencing societal views. Consequently, traditional media increasingly rely on digital platforms for distribution. However, this shift has facilitated the spread of fake news, as malicious actors—motivated by political, economic, social, or health-related goals—intentionally disseminate misleading content. Fake news distorts public opinion, leading to misinformed decisions with far-reaching consequences, including political instability, economic downturns, and social unrest [3]. Unlike reliable news, which promotes informed decision-making, fake news undermines societal trust and exacerbates social problems [4].

Fake news can be defined as intentionally fabricated or manipulated information designed to influence public perception. This misinformation is spread through various formats, including text, images, and videos. Due to its persuasive nature and rapid spread across social media platforms, fake news often gains considerable traction in a short amount of time [5,6]. However, distinguishing fake news from legitimate content is increasingly challenging, especially when misinformation is carefully crafted to appear credible. Given the profound impact of news on society, the need for automated detection systems that can efficiently identify fake news is critical [7]. Manual verification, while useful, is not scalable or efficient enough to cope with the enormous volume of content circulating on social media [8]. This highlights the necessity for real-time, automated detection techniques.

Over the years, several approaches have been proposed to tackle the challenge of fake news detection. Traditional methods mainly relied on machine learning (ML) algorithms, such as Logistic Regression (LR) and Support Vector Machines (SVM), which used predefined features for classification [9,10]. Mehta et al. [11] explored the use of Natural Language Processing (NLP) techniques to extract features from textual data and employed supervised learning models to classify news articles as either fake or real. Despite their utility, traditional methods face significant limitations, particularly in dealing with the vast and dynamic nature of social media content. As a result, the use of deep learning (DL) techniques has gained considerable attention in recent years due to their ability to process large volumes of high-dimensional data more effectively [12,13, 14].

Deep learning, a branch of machine learning, employs multi-layered neural networks to learn complex patterns from data. Among its architectures, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are widely used for fake news detection. While CNNs effectively process textual information, they struggle to capture long-term dependencies between words. RNNs, particularly Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) models, excel in retaining such dependencies, making them well-suited for sequential data like news articles [15]. Despite the success of these models, significant challenges remain in developing systems that can accurately detect fake news at scale. The vast volume of misleading information online and the need for generalizability across topics and languages are key obstacles. In this context, stacking techniques have emerged as a promising approach, aggregating predictions from multiple models to improve classification accuracy.

This study presents a novel fake news detection framework by integrating Kernel Fuzzy Rough Set (KFRS)–based feature selection with a hybrid ensemble of deep and traditional classifiers. The innovation lies in using KFRS to refine feature representations and handle uncertainty in short, noisy texts, while the hybrid ensemble combines LSTM, Bi-LSTM, Logistic Regression, SVM, Passive-Aggressive, and XGBoost classifiers in a two-stage soft voting and stacking scheme. This is an approach not previously applied to fake news detection. Unlike prior studies that relied exclusively on either conventional algorithms or deep neural networks, our approach leverages the complementary strengths of both paradigms. By jointly leveraging TF-IDF and Skip-Gram embeddings, the proposed model captures both statistical and semantic characteristics of news text, overcoming limitations of single-representation approaches. This combination enables the model to refine features under uncertainty. KFRS explicitly models vagueness and overlapping class boundaries. It reduces noise and retains the most discriminative attributes. The framework captures sequential and statistical patterns. By combining LSTM and Bi-LSTM networks with traditional classifiers in a two-stage ensemble (soft voting followed by stacking), it leverages both semantic sequence information and statistical relationships. This improves generalization across datasets. Skip-Gram embeddings are integrated with KFRS to preserve semantic context efficiently. Fine-grained semantic relationships are retained while reducing dimensionality. Both deep learning and traditional classifiers receive high-quality input. Skip-Gram embeddings capture semantic and contextual nuances, ensuring effective integration with KFRS. For example, Xu et al. demonstrated the potential of fuzzy logic for fake news detection [16]. KFRS has also shown strong performance in high-dimensional and multi-class problems, removing irrelevant features while preserving discriminative ones [17,18].

Recent studies have shown that combining traditional machine learning models, such as Logistic Regression (LR), Support Vector Machines (SVM), and XGBoost, with deep learning architectures like LSTM and Bi-LSTM can enhance fake news detection [19,20]. We evaluate our approach on the LIAR, FakeNewsNet, and FakeEdited datasets. Experimental results demonstrate significant improvements in Recall and F1-score, highlighting the model's effectiveness as a reliable solution for automated fake news detection.

The remainder of this paper is organized as follows. Section 1 introduces the research problem, its significance, implications, and motivation. Section 2 reviews related literature, connecting previous works to the present study. Section 3 details data preprocessing and feature extraction, followed by an in-depth discussion of the proposed approach. Section 4 presents the experimental design, dataset specifications, and key results. Section 5 provides an analysis of the computational complexity of the proposed approach. Finally, Section 6 concludes the paper and suggests directions for future research.

## 2. Related Works

In this section, we first provide an overview of machine learning (ML) and deep learning (DL) techniques for fake news detection. We then highlight recent hybrid and ensemble approaches that combine these methods to improve accuracy.

ML has been widely applied for fake news detection due to its ability to classify text based on learned patterns. Classical algorithms, such as Logistic Regression (LR), Support Vector Machines (SVM), and XGBoost, have shown promising results in early studies [21,22]. For example, Mahlous et al. [23] used the TF-IDF technique on Arabic texts to train ML classifiers for detecting COVID-19-related fake news. Similarly, Azad et al. [24] created a Kurdish fake news dataset, which allowed them to train multiple models and achieve an accuracy of 0.88. In another study, Dev et al. [25] analyzed COVID-19-related fake news on Twitter. They combined SVM and Random Forest (RF) classifiers to examine the content. While effective, these methods often require manual feature engineering, which is time-consuming and less scalable for large datasets.

To overcome the limitations of traditional ML, deep learning (DL) methods have been introduced. These models, particularly deep neural networks, can automatically extract features and learn complex patterns from raw data. Alarfaj and Khan [20] evaluated several DL architectures for fake news classification. Similarly, Dutta et al. [26] proposed a DL-based model to classify online articles.

More complex architectures have also been explored. Vo and Phan [13] developed a hybrid model combining Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for Vietnamese fake news detection. A CNN+BiLSTM model trained on the CrisisLexT9 dataset achieved an average accuracy of 0.86 [27]. Furthermore, Almarashy et al. [28] used a Fast Learning Network (FLN) combined with BERT and CNN to enhance detection performance.

Recent studies focus on integrating multiple models to improve prediction accuracy. Singh et al. [19] implemented LSTM, Bi-LSTM, and CNN-BiLSTM models using TensorFlow on datasets including ISOT, TI-CNN, and GRFN. Their preprocessing included stopword removal, stemming, and word vectorization.

In a more advanced approach, Almarashy et al. [29] combined text and image features using BERT, ResNet-

CBAM, and GRU. They then applied feature fusion and dimensionality reduction with Auto-Encoders before final classification via an FLN model.

Ensemble learning has gained attention for its ability to combine multiple classifiers. Ahmad et al. [30] compared various ML and ensemble methods, showing that ensembles consistently outperformed individual models. Luqman et al. [31] integrated NLP techniques with ensemble learning, achieving 0.88 accuracy. Wynne and Swe [32] proposed a framework combining five classifiers to predict news authenticity.

Overall, the literature reveals diverse approaches for fake news detection. Researchers have explored traditional ML, advanced DL, and hybrid strategies. These efforts aim to develop more accurate and scalable models to combat misinformation on online platforms.

## 3. Proposed method

The aim of this study is to develop a novel approach for accurately and efficiently distinguishing between fake and real news. To achieve this, we developed a hybrid framework that combines machine learning, deep learning, natural language processing, and Kernel Fuzzy Rough Set (KFRS) methods for effective fake news classification and credibility assessment. Figure 1 presents an overview of the system's operational workflow, while this section details the proposed model.

### *3.1. Data Preprocessing*

Raw textual data often contain superfluous expressions and meaningless symbols, which must be addressed before analysis [15]. In this study, several preprocessing techniques are applied, including lowercasing, punctuation removal, stop word elimination, tokenization, stemming, and lemmatization [33,34,35]. These steps reduce noise, enhance consistency, and simplify textual data, ensuring that subsequent analyses and models are accurate and efficient [36,37].

### *3.2. Feature extraction*

The feature extraction process is crucial for converting raw textual data into a format suitable for modeling. It encodes text into numerical representations, such as integers or floating-point values. This process enables the creation of feature vectors that machine learning algorithms can process effectively. In this study, six primary textual features are extracted: the news content, the topic, the speaker's identity, the speaker's job title, the state or source of publication, and the political party affiliation of the speaker. These features provide valuable context for distinguishing between fake and real news.

To further enhance detection accuracy, two advanced techniques—TF-IDF (Term Frequency-Inverse Document Frequency) and Word2Vec—are applied. TF-IDF measures the importance of words within a document relative to their frequency across the corpus, helping to identify distinguishing terms. Word2Vec, a neural network-based model, captures semantic relationships by mapping words into dense vector spaces. Together, these techniques reveal underlying patterns and subtle cues, improving the model's ability to differentiate fake news from credible sources [38].

### *3.2.1. Word2Vec Embedding*

Word2Vec is a powerful natural language processing technique that transforms words into numerical vectors in a continuous vector space. This embedding method converts textual data into vector representations, which can be used for tasks such as semantic similarity and information retrieval. Word2Vec includes two algorithms: Continuous Bag of Words (CBOW) and Skip-Gram [39,40].

### *3.2.2. TF-IDF*

Inverse Document Frequency (IDF) is a statistical technique in natural language processing (NLP) that converts text into numerical values. It also measures the relative importance of each word across a set of documents. This method is especially useful for fake news detection. It assesses a word's significance in a document relative to its frequency in other documents of the corpus. IDF is used together with Term Frequency (TF) to provide a more accurate representation of a word's relevance.

The TF-IDF score is calculated using the formula in Equation (1) [38,41], which combines TF and IDF to assign a numerical value representing a word's importance in both the document and the overall corpus.

$$TF - IDF = TF * IDF \tag{1}$$

### *3.3. Proposed Method*

The main contribution of this study is the integration of the Kernel Fuzzy Rough Set (KFRS) framework with a hybrid ensemble combining deep learning and traditional machine learning classifiers. It is structured in three main stages. The proposed hybrid framework processes data through a structured pipeline. First, the raw textual data undergoes preprocessing, including cleaning, tokenization, stop-word removal, and normalization, producing a list of meaningful tokens [15,30,34]. Techniques applied include lowercasing, punctuation removal, stop word elimination, tokenization, stemming, and lemmatization. The dataset is then split into training and test and validation sets immediately after preprocessing, before Word2Vec embedding and KFRS feature selection, ensuring that the model learns exclusively from training data and preventing information leakage.

Next, each token is converted into a numerical representation using Word2Vec embeddings with the Skip-Gram approach and TF-IDF, generating a [sequence_length × 150] matrix for each document. Both TF-IDF and pre-trained Word2Vec embeddings are employed to construct numerical representations. TF-IDF highlights term importance within a document relative to the corpus. Word2Vec, on the other hand, captures semantic relationships and word similarities. Initial experiments comparing TF-IDF and Word2Vec indicated that Word2Vec consistently produces more effective embeddings, particularly in short, noisy, and sparse news texts. The experimental results are presented in Table X. All CNN-based architectures in the proposed framework are trained exclusively on dense Word2Vec embeddings. TF-IDF representations are used solely for baseline evaluation and are not integrated into the deep learning components.

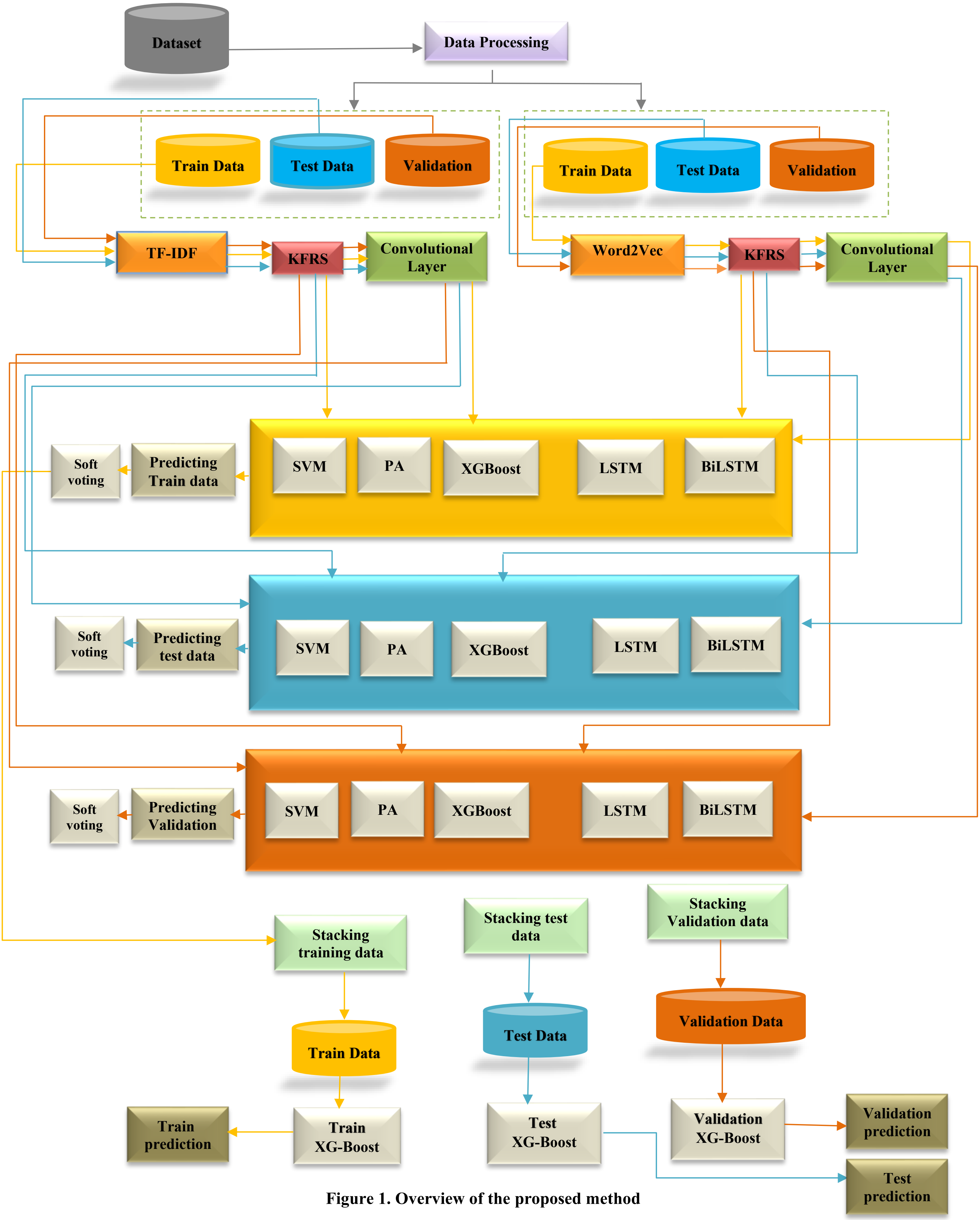

**Figure 1. Overview of the proposed method**

We adopt the Skip-Gram model for constructing embeddings. Skip-Gram embeddings were employed due to their superior ability to capture local semantic relationships in short and noisy texts typical of fake news. GloVe, in contrast, relies on global co-occurrence statistics, and FastText emphasizes subword-level representations. Skip-Gram integrates efficiently with KFRS-based feature selection. This helps retain discriminative semantic features, reduce noise, and maintain computational efficiency.

The short, sparse, and noisy nature of our dataset poses challenges for conventional feature extraction methods such as TF-IDF or Word2Vec, which can produce high-dimensional vectors with irrelevant or noisy dimensions. To reduce noise and preserve the most informative features, Kernel Fuzzy Set (KFRS) feature selection is used, which produces a refined and compact feature matrix. Kernel Fuzzy Set (KFRS) is explicitly integrated as a supervised feature selection and refinement step in the proposed pipeline. After text preprocessing and embedding generation (Word2Vec), the high-dimensional feature vectors are provided as input to the KFRS module. KFRS computes fuzzy similarity relationships between samples using kernel functions and uses class label information to estimate lower and upper approximations of decision classes. Based on these approximations, features are ranked based on their contribution to reducing uncertainty and the boundary regions between fake and real news classes. Only the most informative and distinctive features are retained, while redundant and noisy dimensions are removed. The refined feature matrix produced by KFRS is then used as input for deep learning models (CNN-LSTM and CNN-BiLSTM) and traditional classifiers (SVM, PA, XGBoost). In this way, KFRS acts as a critical middle layer that improves the quality of features before classification, rather than as a standalone preprocessing or post-processing analysis step.

To further assess the contribution of KFRS, we re-executed the proposed hybrid framework without the KFRS stage, using only Word2Vec embeddings for feature representation. This ablation led to a clear decrease in Recall and F1 scores, confirming that KFRS plays a crucial role in enhancing robustness and strengthening the discriminative power of the model. The detailed results are presented in Table IX. All experiments reported in Table IX were performed using 5-fold cross-validation for evaluation. Collectively, these analyses highlight the unique contribution of KFRS within the pipeline and provide empirical justification for adopting Skip-Gram embeddings as the primary representation technique in the proposed model.

In both architectures, CNN–LSTM and CNN–BiLSTM, the convolutional layers are applied before the recurrent layers. These convolutional layers extract local contextual and semantic features from the document embeddings. The resulting feature maps are then fed into the LSTM or Bidirectional LSTM layers, which capture long-term and sequential dependencies. For both architectures, the model input consists of document embeddings with a dimension of [260 × 150], generated using the Skip-Gram Word2Vec model. The convolutional block includes two consecutive 1D convolutional layers, each with 128 filters and a kernel size of 3. The first convolutional layer uses ReLU activation, and the second uses Sigmoid activation. A dropout layer (rate = 0.4) is applied after the convolutional block to prevent overfitting and enhance generalization.

In the CNN–LSTM configuration, the feature maps produced by the convolutional block are sequentially processed by two LSTM layers with 128 and 64 hidden units, respectively. The first LSTM layer uses a ReLU activation, and the second uses a Sigmoid activation. The output of the last recurrent layer is then passed through a Global Max Pooling layer, followed by two Dense layers (with 50 and 1 neurons, respectively) for final classification. The model is trained using the Adam optimizer, with 10 epochs and a batch size of 128.

In the CNN–BiLSTM configuration, the same convolutional block is followed by two Bidirectional LSTM layers. The first BiLSTM layer outputs [None, 260, 256], and the second produces [None, 128], indicating the merged forward and backward activations. Both layers use ReLU and Sigmoid activations for the first and second layers, respectively. Similar to the LSTM model, the outputs are passed through a Global Max Pooling layer and Dense output layers for binary classification. The BiLSTM model is trained using the Adam optimizer, with 10 epochs and a batch size of 64.

In contrast, traditional machine learning classifiers such as SVM, Passive-Aggressive (PA), and XGBoost operate directly on the refined features obtained from KFRS. These models do not require convolutional or recurrent layers, and they learn patterns solely from the compressed feature matrix, allowing efficient and effective classification without sequential modeling.

A two-stage ensemble strategy is then applied. In the first stage, Soft Voting aggregates predictions from all classifiers. In the second stage, Stacking combines these aggregated predictions. Stacking learns optimal weights to correct residual errors, improving overall accuracy and generalization.

The two-stage ensemble—Soft Voting followed by Stacking—leverages complementary strengths of sequence-based (LSTM/Bi-LSTM) and traditional classifiers (SVM, XGBoost, PA). Soft Voting mitigates individual classifier biases, while Stacking learns from residual errors. This results in improved robustness across datasets with varying distributions.

The combination of preprocessing, embedding, KFRS-based feature refinement, and two-stage ensemble learning produces a robust, scalable, and generalizable framework. The model effectively handles noisy, sparse, and short news texts. Cross-dataset evaluation demonstrates consistent performance across LIAR, FakeNewsNet, and FakeEdited. Low variance in Accuracy, Recall, and F1 across these datasets confirms the framework’s stability and its ability to capture the underlying patterns of fake news beyond dataset-specific characteristics. Overall, this integrated approach ensures accurate classification of news as fake or non-fake, while maximizing robustness and discriminative power. The experimental results are presented in Table V.

### *3.3.1. KFRS Integration for Feature Improvement*

To enhance the robustness and generalization of the proposed hybrid model for fake news detection, a feature evaluation mechanism based on the Kernel Fuzzy Rough Set (KFRS) is integrated into the pipeline. Kernel Fuzzy Rough Set (KFRS) is employed for feature selection due to its ability to handle uncertainty, vagueness, and overlapping class boundaries in high-dimensional spaces. Unlike conventional methods such as Chi-square, mutual information, RFE, and PCA, KFRS leverages class labels and fuzzy similarity relations to select discriminative, noise-robust features. This is especially useful for fake news detection, where features are sparse and noisy, helping retain the most informative and relevant ones for the hybrid framework. Let $x_i$ and $x_j$ be two vectorized samples from Word2Vec or TF-IDF. Suppose U is a finite set of samples. $x_i \in U$ represented by a vector $x_{i1} + x_{i2} + \cdots + x_{in} \in R^n$. Therefore, U can be considered as a subset of $R^n$ . According to equation (2), the fuzzy similarity between two samples is computed with Gaussian kernel function, where $\|x_i - x_j\|^2$ the input is the Euclidean distance between the samples:

$$k(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2 / 2\delta^2\right) \tag{2}$$

This kernel matrix defines a fuzzy T-equivalence relation that satisfies reflexivity, symmetry, and Tcos-transitivity. As shown in Equations (3) and (4), the fuzzy lower approximation of a set X is first computed, capturing the elements that definitely belong to the concept under consideration. Subsequently, the fuzzy upper approximation is derived to include elements that possibly belong to the concept.

$$\underline{R_{\max}} X(x) = \inf_{y \in U} \max(1 - R(x,y)X(y)) \tag{3}$$

$$\overline{R_{\min}} X(x) = \sup_{y \in U} \min(R(x,y)X(y)) \tag{4}$$

Here, max and min are used as the fuzzy upper and lower approximation operators, respectively. The boundary region, computed as the difference between upper and lower approximations, represents the degree of uncertainty surrounding each feature. Features that contribute to a smaller boundary region are considered more significant and are selected for subsequent classification.

### *3.3.2. Combining predictions with the Stacking algorithm*

The predictions generated by the five base classifiers were combined using the Stacking algorithm and Soft voting . This process creates a new dataset, which serves as input for the final classifier. The final classifier is selected based on the evaluation of the outputs from the machine learning (ML) and deep learning (DL) models used in the previous step. This approach ensures optimal accuracy and system performance by leveraging the predictive strengths of all the classifiers. Prior to generating the final prediction, the parameters and variables of the classifiers are meticulously tuned. The pseudocode for this process is provided in Algorithm I.

### *3.3.3. structure of the proposed model*

The overall architecture operates in two stages. In the first stage, the CNN–LSTM and CNN–BiLSTM configurations extract high-level contextual features, and the corresponding mathematical formulations are presented in the following section. In the second stage, KFRS-refined features are directly classified using traditional machine learning algorithms, eliminating the need for convolutional processing. To facilitate understanding, an illustrative example is provided. Consider a news document written in English containing D words. This document serves as the initial input to the proposed architecture. According to Equation (5), the input text is first represented as a series of word vectors using embedding techniques.

$$x_{1:n} = x_1 + x_2 + \cdots + x_n \tag{5}$$

**Algorithm I.** Pseudocode for stacking classifier

```
Stack predictions with dataset
Input: prediction of all base classifiers
Output: stacked train and test data

stacked_X_train = np.hstack((X_train,
    svm_train_valid_predictions.reshape((-1, 1)),
    xgb_train_valid_predictions.reshape((-1, 1)),
    lstm_train_valid_predictions.reshape((-1, 1)),
    bilstm_train_predictions.reshape((-1, 1)),
    pa_train_valid_predictions.reshape((-1, 1))))

stacked_X_test = np.hstack((X_test,
    svm_predictions.reshape((-1, 1)),
    xgb_predictions.reshape((-1, 1)),
    lstm_predictions.reshape((-1, 1)),
    bilstm_predictions.reshape((-1, 1)),
    pa_predictions.reshape((-1, 1))))
```

In equation (5), $i$ represents the position of the word, as well as the dimensions and length of the word vector. In equation (6), $x \in R^{L\times d}$ is an input which $x_i \in R^d$ refers to a word in the news document $x$. In the CNN–LSTM and CNN–BiLSTM configurations, sequences of consecutive word embeddings are combined and passed into the convolutional layer. In the convolutional layer, $n \in R^{k\times d}$ is the vector, and the kernel $f$ represents the operation performed by the convolutional layer. In this layer, the kernel $f$, which is a sigmoid function, moves across the words, analyzing them from $(j + k - 1)$ and extracts local features of the text.

$$w_i = [x_i + x_{i+j} + \ldots + x_{j+k-1}] \tag{6}$$

$$c_i = f(w.x_{i:i+h-1} + b) \tag{7}$$

In equation (6), $k$ represents the length of the kernel, whereas in equation (7), the terms $b \in R$ and $c \in R^{n-h+1}$ denote the element-wise product of mathematical operations. All features extracted from $w_i$ are mapped to the features of $c$, and the value of $c$ is calculated based on formula (8):

$$c = [c_1, c_2, \dots, c_{n-h+1}] \tag{8}$$

Equation (8) aggregates all the feature maps generated in the previous step into a unified representation vector and preserving important local dependencies identified by the convolution filters. After the convolution operation, Max Pooling is applied to retain the most informative features while reducing dimensionality. This results in a compact feature vector that captures the most salient local patterns in the text. Moreover, Max Pooling mitigates noise and overfitting by preserving only the strongest activation , which are mathematically defined in equation (9).

$$c' = max\{c_1 , c_2 , \dots , c_{n-h+1}\} \tag{9}$$

After obtaining the final feature vector from the CNN and Max Pooling layers, sequential modeling is performed using LSTM and Bi-LSTM to capture temporal dependencies and contextual relationships that CNN alone cannot model. Let c^\prime denote the feature vector extracted from CNN (Max Pooling output), which serves as input to the LSTM layer. The LSTM employs gating mechanisms to control the flow of information over time. It combines the current input with the previous hidden state through three fully connected layers. Each layer uses a sigmoid activation function to form one of the gates: input, forget, or output. The outputs of these gates are constrained within the range [0,1], regulating how much information is stored, forgotten, or passed to the next step. Additionally, a tanh-activated node generates the candidate cell state, allowing the network to selectively retain and propagate relevant information over time. Numerically, for a hidden unit "h", a time step "n", and an input "g", the gates at time t are computed according to Equations (10), (11) and (12).

$$\text{Input Gate} = I_t = \sigma\ (x_t * W_{xi} + h_{t-1} * W_{hi} + b_i) \tag{10}$$

$$\text{Output Gate} = O_t = \sigma\ (x_t * W_{xo} + h_{t-1} * W_{ho} + b_o) \tag{11}$$

$$\text{Forget Gate} = F_t = \sigma\ (x_t * W_{xf} + h_{t-1} * W_{hf} + b_f) \tag{12}$$

Where $b_i, b_o, b_f \in R^{1\times h}$ are biases and $W_{xi}, W_{xo}, W_{xf} \in R^{g\times h}$ and $W_{hi}, W_{ho}, W_{hf} \in R^{h\times h}$ are weights.

The LSTM layer captures the sequential patterns of feature vectors extracted by the CNN, effectively modeling contextual dependencies across the document. To further enhance this representation, a Bidirectional LSTM (Bi-LSTM) layer is employed, enabling the model to process the sequence in both forward and backward directions. This bidirectional structure produces hidden states that integrate information from both preceding and succeeding tokens, leading to a more comprehensive contextual understanding. The final output $y$ is derived by combining the representations from both directions. The governing equations for the Bi-LSTM are defined as follows:

$$H_{i1} = f\ (U_1 * x_i + W_1 * h_{i-1}) \tag{13}$$

$$H_{i2} = f\ (U_2 * x_i + W_2 * h_{i-1}) \tag{14}$$

$$y_i = \sigma\ (V * [h_{i1} * h_{i2}]) \tag{15}$$

In this context $[h_{i1} * h_{i2}]$ represents the hidden state at time step within the forward LSTM layer, and f denotes the activation function, such as ReLU or tanh, applied to the weighted sums. $U_1$ and $U_2$ are the weight matrices connecting the input to the hidden units in the forward LSTM, while $W_1$ and $W_2$ correspond to the hidden-to-hidden connection weights. $V$ is the weight matrix used to map the concatenated hidden states from both the forward and backward LSTM layers. Finally, the output from the Bi-LSTM layer is fed into a dense layer followed by a Sigmoid activation to produce the final probability distribution over the classes (fake or real news):

$$P(i|t; \emptyset) = \frac{exp(y_{ij})}{\sum_{k=1}^{n} exp\ (y_{ij})} \tag{16}$$

The resulting $P(i|t; \emptyset)$ represents the predicted probability used for the final classification This combination allows the model to capture both local features and contextual dependencies within the text. As a result, the system effectively learns semantic patterns and temporal relationships that are essential for accurate fake news detection. This step classifies the news document as either fake or real based on the combined local and sequential features extracted by the model. Table I illustrates the layered architecture of the proposed model, including the values for both the input and output layers.

### *3.4. Real-World Applications and Deployment Scenarios*

The proposed hybrid fake news detection framework is designed for practical deployment. It can assist human reviewers in fact-checking platforms by prioritizing potentially misleading content, reducing manual workload while keeping critical decisions under human supervision. It can also be integrated into social media moderation systems, flagging suspicious news early. Its modular pipeline—preprocessing, embedding generation, KFRS-based feature refinement, and classification—allows easy integration with existing infrastructures. KFRS-based feature selection improves computational efficiency by reducing redundancy and noise, making the system suitable for large-scale streaming data on cloud or on-premise platforms.

## 4. Experimental results

This section discusses the key aspects involved in the construction and evaluation of the ensemble classifier, which results from combining machine learning (ML) and deep learning (DL) classifiers, with the goal of classifying news as either fake or real.

### *4.1. Data sets*

In this study, three benchmark datasets were employed to evaluate the performance of the proposed hybrid model. Each dataset was divided into three subsets: 60% for training, 20% for validation, and 20% for testing. The specific data split configuration is summarized in Table II.

The LIAR dataset, introduced by Wang [42], consists of 12,836 short statements accompanied by six associated metadata features, specifically curated for the task of fake news detection. It has been publicly available since January 20, 2022. The dataset includes news statements collected from various sources such as TV advertisements, Facebook posts, Twitter feeds, interviews, and political debates. These diverse origins provide a broad representation of online and media-based misinformation. Each statement in the dataset is manually annotated as either fake or real. The FakeNewsNet dataset, developed by Shu et al. contains 15,500 posts from 32 pages (14 related to conspiracy theories and 18 scientific pages), accumulating over 23,000 likes. It includes news content, social context, and temporal-spatial information, designed for fake news studies on social media. For this study, the textual content and labels were used for binary classification.

The FakeEdited dataset, introduced by Nakamura et al. is a multimodal dataset containing over 1 million fake news samples across various categories. It includes text, images, metadata, and comments, specifically designed for high-accuracy fake news detection.

**Table I.** Layered LSTM & BILSTM architecture

| Algorithm | Layer (type) | Output Shape |
|---|---|---|
| LSTM | Embeddings (Embedding) | (None, 260, 150) |
| | lstm_layer1 (LSTM) | (None, 260, 128) |
| | lstm_layer2 (LSTM) | (None, 260, 64) |
| | (GlobalMax-Pooling1D) | (None, 64) |
| | dense_2 (Dense) | (None, 50) |
| | dense_3 (Dense) | (None, 1) |
| BILSTM | Embeddings (Embedding) | (None, 260, 150) |
| | Bidirectional (Bidirectional) | (None, 260, 256) |
| | Bidirectional_1 (Bidirectional) | (None, 128) |

*4.2. Parameter Setting*

Parameter tuning is crucial for controlling model behavior and preventing overfitting. In this study, the Grid Search technique was employed to determine the optimal parameter values and improve system performance. Grid Search performs an exhaustive search over the predefined parameter space to identify the best combination for each variable. The algorithm evaluates all possible combinations of the selected values on the preprocessed data, selecting the configuration that yields the highest model performance. The parameters tuned in this study are summarized in Table III.

**Table II**. the number of samples in the LIAR dataset

| | Dataset Name | Number of Instances |
|---|---|---|
| LIAR | Train | 10,240 |
| | Validation | 1,284 |
| | Test | 1,267 |
| | Total | 12,791 |
| FakeNewsNet | Train | 918,13 |
| | Validation | 639,4 |
| | Test | 639,4 |
| | Total | 23,196 |
| FakeEdited | Train | 637,864 |
| | Validation | 212,621 |
| | Test | 212,621 |
| | Total | 1,063,106 |

**Table III.** Parameter setting

| Hyper Parameter Tuning ML & DL | | |
|---|---|---|
| Models | 2 class | 6 class |
| SVM | C = 10<br>Gamma = scale<br>Kernel = sigmoid | Kernel = poly<br>decision-Function = ovr |
| XGBOOST | Max depth = 20<br>Min child weight = 1<br>Eta = 0.0<br>Subsample = 0.2<br>colsample_bytree = 0.2 | Multi_softmax |
| PA | Iteration =100<br>C = 0.1 | |
| LSTM | Activation Function_layer1 = Relu<br>Activation Function_layer2 = Sigmoid<br>Optimizer = Adam<br>Epoch = 10<br>Batch size = 128<br>Dropout = 0.4 | |
| BILSTM | Activation Function_layer1 = Relu<br>Activation Function_layer2 = Sigmoid<br>Optimizer = Adam<br>Epoch = 10<br>Batch size = 64<br>Dropout = 0.4 | |

### *4.3. Reducing Classes*

The dataset originally contains six news categories: False (completely false), Barely-true (barely true), Half-true (half true), Mostly-true (mostly true), True (completely true), and Pants-on-fire (completely false). Experiments were conducted in two phases: first, using all six classes, and second, consolidating the dataset into two classes to simplify the problem and enhance performance.

To reduce complexity, the Half-true class was removed, as its ambiguous nature could increase error rates. Subsequently, the True and Mostly-true classes were grouped as real news, while the False, Barely-true, and Pants-on-fire classes were grouped as fake news.

### *4.4. Evaluation of metrics*

The proposed method was implemented in a Windows operating system environment using the Python programming language and Jupyter Notebook software. To evaluate the performance and accuracy of the proposed method, four standard metrics Accuracy, Recall, Precision, and F1-Score were utilized. These metrics are commonly used to assess the effectiveness of classification systems: TP (True Positive) – fake news correctly classified as fake; TN (True Negative) – real news correctly classified as real; FP (False Positive) – fake news incorrectly classified as real; and FN (False Negative) – real news incorrectly classified as fake. The formulas for calculating the aforementioned metrics are as follows (Equations 17 to 20):

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$F1 - score = 2 \times \frac{prec \times recall}{prec + recall} \tag{20}$$

The results of each classification algorithm, including performance on both training and validation sets with k-Fold = 5, are summarized in Table IV. To provide a detailed assessment of the final classifiers, the Confusion Matrix is presented for each dataset. Figure 2 illustrates the number of correctly and incorrectly classified instances across all datasets, offering a precise and comprehensive evaluation of model accuracy and robustness.

In table VI for LIAR dataset and the SVM classifier, the parameter C, which controls the model's regularization and allowable error margin, was tested with values of 1, 5, and 10. Grid Search identified C = 10 as optimal. The gamma parameter was evaluated using the options scale and auto, with scale selected to prevent excessive sensitivity to the features. A sigmoid kernel was employed. Using 5-Fold Cross-Validation, the SVM achieved an overall accuracy of 0.77.

For the Passive-Aggressive (PA) algorithm, after tuning, C was set to 0.1 and the number of epochs to 100. The model obtained a maximum 5-Fold Precision of 0.82 (see Table IV).

The XGBoost classifier, a gradient-boosted decision tree algorithm, was optimized by adjusting key parameters. The subsample and colsample_bytree were set to 0.2, max_depth to 20, and the learning rate (eta) to 0.01 based on performance evaluation. The classifier achieved a maximum Recall of 0.87 and a minimum Precision of 0.73, as summarized in Table IV.

**Table IV.** Results in 2-class – LIAR 5-fold

| Algorithms | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| SVM | 0.73 | 0.73 | 0.73 | 0.77 |
| PA | 0.75 | 0.72 | 0.82 | 0.78 |
| XGBOOST | 0.85 | 0.87 | 0.73 | 0.82 |
| LSTM | 0.83 | 0.86 | 0.75 | 0.76 |
| BI-LSTM | 0.80 | 0.84 | 0.69 | 0.75 |
| Results in 2-class – FakeNewsNet 5-fold | | | | |
| Algorithms | F1 | Recall | Precision | Accuracy |
| SVM | 0.80 | 0.80 | 0.81 | 0.83 |
| PA | 0.82 | 0.81 | 0.88 | 0.85 |
| XGBOOST | 0.92 | 0.93 | 0.89 | 0.91 |
| LSTM | 0.89 | 0.91 | 0.89 | 0.91 |
| BI-LSTM | 0.87 | 0.90 | 0.78 | 0.82 |
| Results in 2-class – FakeEdited 5-fold | | | | |
| Algorithms | F1 | Recall | Precision | Accuracy |
| SVM | 0.79 | 0.77 | 0.80 | 0.80 |
| PA | 0.78 | 0.78 | 0.85 | 0.83 |
| LSTM | 0.90 | 0.89 | 0.88 | 0.89 |
| BI-LSTM | 0.88 | 0.88 | 0.77 | 0.80 |

The proposed LSTM model consists of six layers (Table I). The Embedding layer converts words into numerical vectors using Word2Vec, with an input length of 260 and a vector dimension of 150. Two LSTM layers, with 128 and 64 neurons respectively, capture sequential dependencies in the data. A GlobalMaxPool1D layer reduces feature dimensionality, followed by fully connected Dense layers with ReLU and Sigmoid activations for binary classification. The model was trained using the Adam optimizer with binary cross-entropy loss, 10 epochs, a batch size of 128, and a Dropout of 0.4 to prevent overfitting. This LSTM network achieved an average Recall of 0.86.

The Bi-LSTM classifier was implemented as a five-layer network, as detailed in Algorithm 1. The Embedding layer converts words into numerical vectors, with an input length of 260, a vector dimension of 150, and a vocabulary size of 8,636. Two subsequent LSTM layers, containing 128 and 64 neurons respectively, capture semantic relationships and key features from the data. The final Dense layers employ ReLU and Sigmoid activation functions. The model was trained for 10 epochs with a batch size of 64.

As reported in Table IV, the Bi-LSTM achieved an average Recall of 0.84. Among the base classifiers, XGBoost demonstrated the best performance. The final XGBoost model was constructed using Stacking, which takes the outputs of five base classifiers along with the original training data as input, thereby integrating previous predictions into the training process.

After hyperparameter tuning, the learning rate was set to 0.01, while the maximum tree depth, row subsample, and column subsample were set to 2, 0.2, and 0.8, respectively. The Confusion Matrix of the final classifier is shown in Figure 2, indicating that 183 fraudulent and 399 non-fraudulent samples were correctly classified, whereas 370 fraudulent and 50 non-fraudulent samples were misclassified.

Figure 3 presents the evaluation results of each classifier based on the selected performance metrics. Among the models, the Passive-Aggressive (PA) classifier achieved the highest Precision, whereas the Bi-LSTM exhibited the lowest Accuracy. In terms of Recall and F-Score, the XGBoost algorithm outperformed all other classifiers.

Within the deep learning models, LSTM demonstrated superior performance compared to Bi-LSTM. As summarized in Table IV, XGBoost achieved the best overall results among all tested classifiers and was therefore selected as the final model.

The proposed model was further evaluated under two scenarios: binary classification (two classes) and multi-class classification (six classes), to assess its robustness and generalizability across different classification settings.

For a fair comparison, the methods in [19], [31], and [27] were re-implemented and evaluated on the LIAR, FakeNewsNet, and FakeEdited datasets under identical conditions. Due to limited access to the original datasets, their models could not be assessed on the original data. Hence, all methods, including the proposed hybrid model, were tested using the same training, validation, and testing splits. The comparative results are summarized in Table V. As shown, the proposed model outperforms the others across all metrics. These gains are attributed to advanced preprocessing, optimized feature extraction, and the ensemble-based design. Evaluating all methods under the same conditions ensures a fair comparison and accurately reflects the effectiveness of the proposed approach.

For the LIAR dataset, the system achieved an Accuracy of 0.88, an F1-score of 0.90, a Precision of 0.88, and a Recall of 0.89. As shown in Table VI, the highest average value is attributed to the XGBoost classification algorithm. For the FakeNewsNet dataset, the proposed method achieved an Accuracy of 0.91, a Precision of 0.90, a Recall of 0.92, and an F1-score of 0.92, demonstrating strong performance across this dataset. For the FakeEdited dataset, the proposed method achieved better performance, with a Recall of 0.95, an F1-score of 0.94, a Precision of 0.93, and an Accuracy of 0.93. These results indicate that the system effectively handles this dataset, outperforming the other classifiers. Moreover, the evaluation metrics for each classifier on the six-class data are presented in Table VI.

The decrease in performance in the six-class setting can be explained by several data-related factors. First, some LIAR classes (e.g., True, Mostly True, Half True) have overlapping meanings, which makes distinguishing them challenging even for humans. Second, there is a class imbalance problem, as categories like Pants-on-Fire contain far fewer samples than others. Moreover, the short and noisy nature of LIAR statements limits the available contextual information, making fine-grained classification inherently difficult. In contrast, the proposed model maintains strong and stable performance on the binary classification tasks of FakeNewsNet and FakeEdited, suggesting that the observed decline in six-class performance is mainly due to dataset characteristics rather than model weakness.

**Table V**. Final results with test data with 2 classes 5-fold

| [31] | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| LIAR | 0.85 | 0.85 | 0.87 | 0.86 |
| FakeNewsNet | 0.87 | 0.87 | 0.86 | 0.986 |
| FakeEdited | 0.88 | 0.88 | 0.89 | 0.87 |
| Std | 0.005 | 0.005 | 0.005 | 0.005 |

| [27] | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| LIAR | 0.83 | 0.84 | 0.82 | 0.83 |
| FakeNewsNet | 0.89 | 0.90 | 0.89 | 0.90 |
| FakeEdited | 0.91 | 0.92 | 0.90 | 0.90 |
| Std | 0.003 | 0.003 | 0.003 | 0.003 |

| [19] | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| LIAR | 0.86 | 0.87 | 0.85 | 0.85 |
| FakeNewsNet | 0.88 | 0.88 | 0.86 | 0.87 |
| FakeEdited | 0.90 | 0.90 | 0.89 | 0.89 |
| Std | 0.003 | 0.003 | 0.003 | 0.003 |

| **Proposed method** | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| LIAR | 0.90 | 0.89 | 0.88 | 0.88 |
| FakeNewsNet | 0.92 | 0.92 | 0.90 | 0.91 |
| FakeEdited | 0.94 | 0.95 | 0.93 | 0.93 |
| Std | 0.004 | 0.004 | 0.004 | 0.004 |

**Table VI**. Final results with test data with 6 classes

| Algorithms | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| SVM | 0.86 | 0.86 | 0.84 | 0.85 |
| PA | 0.86 | 0.85 | 0.87 | 0.88 |
| XGBOOST | 0.88 | 0.86 | 0.86 | 0.88 |
| LSTM | 0.87 | 0.87 | 0.85 | 0.86 |
| BI-LSTM | 0.87 | 0.87 | 0.85 | 0.86 |

To thoroughly evaluate the robustness and generalizability of our proposed hybrid model, we evaluated its performance on the hold-out test set of each dataset, ensuring assessment on completely unseen data. Table VII presents the mean of Accuracy, Precision, Recall, and F1-score across the CV folds, along with the results on the hold-out test sets. For example, on the hold-out test set of FakeNewsNet, the F1-score was 0.90, and the Recall was 0.91. The low standard deviation across CV folds indicates stable performance, while the hold-out results confirm strong generalization across datasets. These findings demonstrate that, although the initial training performance suggested potential overfitting, the final reported results are reliable. The combination of cross-validation, hold-out evaluation, and testing on multiple datasets confirms that the model's performance is robust and stable. In other words, the extremely high F1 and Recall scores reported for LIAR are not artifacts of overfitting. They reflect the model's true ability to effectively detect fake news in real-world scenarios.

**Table VII.** Results Hold-out with 2 classes 5-fold

| Dataset | Metric | mean | Hold-out |
|---|---|---|---|
| LIAR | Accuracy | 0.88 | 0.86 |
| | Precision | 0.88 | 0.87 |
| | Recall | 0.89 | 0.88 |
| | F1-score | 0.90 | 0.88 |
| FakeNewsNet | Accuracy | 0.91 | 0.90 |
| | Precision | 0.90 | 0.89 |
| | Recall | 0.92 | 0.91 |
| | F1-score | 0.92 | 0.90 |
| FakeEdited | Accuracy | 0.93 | 0.92 |
| | Precision | 0.93 | 0.91 |
| | Recall | 0.95 | 0.92 |
| | F1-score | 0.94 | 0.92 |

**Table VIII.** Results BERT and RoBER with 2 classes 5-fold

| | F1 | Recall | Precision | Accuracy |
|---|---|---|---|---|
| Proposed method | | | | |
| LIAR | 0.90 | 0.89 | 0.88 | 0.88 |
| FakeNewsNet | 0.92 | 0.92 | 0.90 | 0.91 |
| FakeEdited | 0.94 | 0.95 | 0.93 | 0.93 |
| BERT | | | | |
| LIAR | 0.86 | 0.85 | 0.84 | 0.84 |
| FakeNewsNet | 0.88 | 0.88 | 0.90 | 0.89 |
| FakeEdited | 0.91 | 0.92 | 0.90 | 0.90 |
| RoBER | | | | |
| LIAR | 0.88 | 0.88 | 0.86 | 0.87 |
| FakeNewsNet | 0.89 | 0.90 | 0.89 | 0.90 |
| FakeEdited | 0.92 | 0.91 | 0.90 | 0.90 |

**Table IX.** Results With KFRS and Without KFRS

| Proposed method | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| With KFRS | | | | |
| LIAR | 0.88 | 0.88 | 0.89 | 0.90 |
| FakeNewsNet | 0.91 | 0.90 | 0.92 | 0.92 |
| FakeEdited | 0.93 | 0.93 | 0.95 | 0.94 |
| Without KFRS | | | | |
| LIAR | 0.84 | 0.84 | 0.86 | 0.86 |
| FakeNewsNet | 0.88 | 0.87 | 0.90 | 0.88 |
| FakeEdited | 0.89 | 0.89 | 0.90 | 0.91 |

To evaluate the balance between model complexity and predictive performance, we conducted a set of experiments comparing the proposed ensemble model with simpler baseline approaches. we compared the proposed ensemble model with simpler architectures such as BERT and RoBERTa (see Table VIII). All models were trained and tested on the same dataset using identical preprocessing steps to ensure fairness. The evaluation metrics included Accuracy and F1-Score, while training and inference times were also recorded to assess computational cost. The results demonstrate that, despite requiring more computation, the ensemble consistently outperforms other models in terms of accuracy and robustness across all datasets, thereby justifying its added complexity.

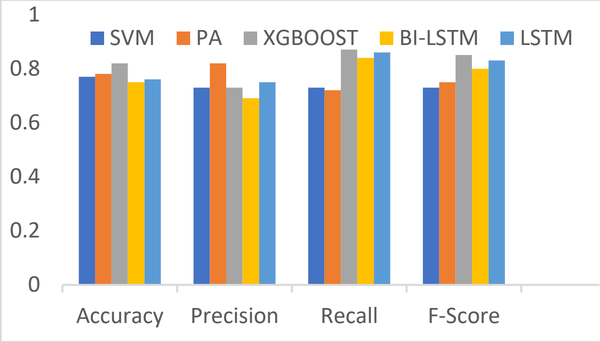

Figure 3. Comparison of evaluation criteria of basic classifiers

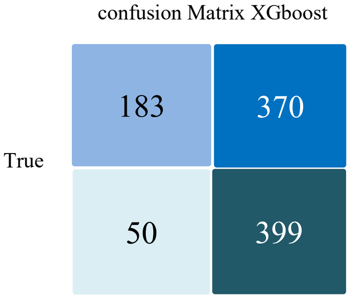

Figure 2. Final XGBoost classifier confusion matrix

**Table X.** Results Word2Vec -LIAR

| | | |
|---|---|---|
| SVM | Accuracy | 0.75 |
| | Precision | 0.77 |
| | Recall | 0.82 |
| | F-Score | 0.86 |
| | | |
| XGBoost | Accuracy | 0.76 |
| | Precision | 0.75 |
| | Recall | 0.90 |
| | F-Score | 0.91 |
| | | |
| PA | Accuracy | 0.74 |
| | Precision | 0.78 |
| | Recall | 0.76 |
| | F-Score | 0.72 |
| | | |
| LSTM | Accuracy | 0.79 |
| | Precision | 0.78 |
| | Recall | 0.96 |
| | F-Score | 0.82 |
| | | |
| BiLSTM | Accuracy | 0.76 |
| | Precision | 0.71 |
| | Recall | 0.89 |
| | F-Score | 0.82 |

**Table X.** Results TTF-IDF-LIAR

| | | |
|---|---|---|
| SVM | Accuracy | 0.73 |
| | Precision | 0.72 |
| | Recall | 0.82 |
| | F-Score | 0.73 |
| | | |
| XGBoost | Accuracy | 0.61 |
| | Precision | 0.69 |
| | Recall | 0.70 |
| | F-Score | 0.73 |
| | | |
| PA | Accuracy | 0.70 |
| | Precision | 0.70 |
| | Recall | 0.86 |
| | F-Score | 0.72 |
| | | |
| LSTM | Accuracy | 0.76 |
| | Precision | 0.65 |
| | Recall | 0.85 |
| | F-Score | 0.79 |
| | | |
| BiLSTM | Accuracy | 0.64 |
| | Precision | 0.61 |
| | Recall | 0.75 |
| | F-Score | 0.60 |

## 5. Computational complexity

The computational complexity of the proposed hybrid model primarily arises from the integration of multiple feature extraction methods, namely TF-IDF and Word2Vec, along with several classifiers. Computing TF-IDF for a corpus of $N$documents with a vocabulary size of $V$has a complexity of $O(N \times V)$. For Word2Vec using the Skip-Gram model, the per-epoch complexity is $O(E \times T \times d \times \log(V))$ , where $E$ is the number of training epochs, $T$is the total number of words, $d$is the embedding dimension, and $V$represents the vocabulary size. Sequence modeling with LSTM or Bi-LSTM layers incurs a computational cost per sample of $O(T \times d \times h + h^2)$, with $T$ being the sequence length, $d$ the input feature dimension, and $h$the number of hidden units. The stacking classifier adds further complexity depending on the number and type of base classifiers; however, this cost can be mitigated through parallel computation. Overall, while the hybrid architecture introduces moderate computational overhead, the use of KFRS-based feature selection and parallelized training ensures that the model maintains high accuracy and robustness without imposing prohibitive time or hardware requirements. Despite its computational complexity, the method exhibits strong performance and consistent stability across the three datasets.

## 6. Conclusion

In this study, the implementation and evaluation of the proposed model for fake news detection were thoroughly investigated using the LIAR, FakeEdited, and FakeNewsNet datasets. The process began with data preprocessing, which included data cleaning and labeling, stopword removal, and performing layering and vectorization. To simplify the classification task, the number of classes in the dataset was reduced from six to two. However, experiments were also conducted using the original six-class configuration to evaluate the model's performance in more complex conditions.

In addition to the raw news text, several text-based features were included in the data to identify more accurate patterns. To build the model, a combination of machine learning models—Support Vector Machine (SVM), Passive Aggressive (PA), and XG-Boost—and deep learning models such as Long Short-Term Memory (LSTM) and Bi-LSTM (Bi-LSTM) were employed. The final prediction was generated using a soft voting group approach. In the two-class classification scenario, the model achieved an F1 score of 0.90 and an accuracy of 0.88, indicating its strong ability to distinguish fake news from real news.

After that, a clustering approach was applied to combine the outputs of the base classifiers with the original input features. The resulting dataset was then used to train the final classifier implemented using the XG-Boost model. To ensure the robustness of the findings, several other learning models were trained and evaluated on the preprocessed LIAR dataset. Their performance was compared using standard evaluation metrics, including precision, accuracy, recall, and F1 score. A comparison between the results of this study and those reported in previous works [19], [27], and [31] shows that the proposed model achieves superior performance. This improvement can be attributed to several factors, including the use of additional features, the reduction in the number of output classes, and the integration of diverse classification models. In the two-class classification setting, the model showed excellent recall and effectively identified fake news samples, although there is still room for improvement in accuracy. The observed performance degradation in the six-class classification can be explained by the increased data complexity and feature overlap between classes.

Moreover, this study integrated the Kernel Fuzzy Rough Set (KFRS) method to enhance feature selection and improve classification accuracy. The inclusion of KFRS enabled more effective handling of uncertainty and vagueness in the data, resulting in greater model robustness and performance compared to conventional feature selection techniques. This approach was particularly beneficial for reducing dimensionality while preserving essential discriminative information, thereby supporting more accurate fake news detection. The evaluation conducted on real-world datasets further demonstrated the model's strong performance and its ability to deliver consistent and reliable results. For future work, we plan to incorporate user behavior patterns into automated fake news detection systems and develop an online framework capable of providing flexible and real-time predictions. Additionally, we aim to combine N-Gram-based data representation with both Word2Vec and TF-IDF techniques to further enhance feature extraction.

The proposed hybrid framework for fake news detection offers several potential real-world applications. It can be integrated into online news platforms and media outlets to automatically flag or filter misleading content, thereby reducing the spread of misinformation. Social media platforms such as Twitter and Facebook can deploy the model in real time to assess the credibility of posts and alerts. The framework can also be used offline for analytical purposes, including trend monitoring and historical news analysis [43] The model demonstrates high precision, recall, and F1-scores, particularly in detecting short and noisy news items, which makes it suitable for real-time evaluation. However, for long and complex articles, additional tuning or hierarchical feature extraction may be required to maintain optimal performance. From both social and commercial perspectives, this framework can enhance public trust in media, assist companies in monitoring user-generated content, and support government and non-government organizations in mitigating misinformation campaigns. Its adaptability to multiple datasets—LIAR, FakeNewsNet, and FakeEdited —further confirms its strong generalizability, making it suitable for deployment across diverse domains and languages [44].

References

[1] D.G. Dev, V. Bhatnagar, B.S. Bhati, M. Gupta, A. Nanthaamornphong, “LSTMCNN: A hybrid machine learning model to unmask fake news”, Heliyon, vol. 10, no. 3, 2024.

[2] N. Aslam, I. Ullah Khan, F.S. Alotaibi, L.A. Aldaej, A.K. Aldubaikil, “Fake detect: A deep learning ensemble model for fake news detection”, Complexity, vol. 2021, pp. 1–8, 2021.

[3] G. De Magistris, S. Russo, P. Roma, J.T. Starczewski, C. Napoli, “An explainable fake news detector based on named entity recognition and stance classification applied to COVID-19”, Information, vol. 13, no. 3, p. 137, 2022.

[4] A.A. Obaid, H. Khotanlou, M. Mansoorizadeh, D. Zabihzadeh, “Multimodal fake-news recognition using ensemble of deep learners”, Entropy, vol. 24, no. 9, 2022.

[5] E. Hashmi, S.Y. Yayilgan, M.M. Yamin, S. Ali, M. Abomhara, “Advancing fake news detection: Hybrid deep learning with fasttext and explainable AI”, IEEE Access, 2024.

[6] A.M. Ali, F.A. Ghaleb, B.A.S. Al-Rimy, F.J. Alsolami, A.I. Khan, “Deep ensemble fake news detection model using sequential deep learning technique”, Sensors, vol. 22, no. 18, p. 6970, 2022.

[7] Q. Chang, X. Li, Z. Duan, “Graph global attention network with memory: A deep learning approach for fake news detection”, Neural Networks, vol. 172, p. 106115, 2024.

[8] Q. Zhang, Z. Guo, Y. Zhu, P. Vijayakumar, A. Castiglione, B.B. Gupta, “A deep learning-based fast fake news detection model for cyber-physical social services”, Pattern Recognition Letters, vol. 168, pp. 31-38, 2023.

[۹] فریما مقدم، پیوند قادریان، موسی شمسی، "تشخیص اختلال نقص‌توجه بیش‌فعالی با استفاده از تحلیل ارتباط نواحی مختلف مغز و روش تاب‌خوردگی زمانی پویا"، مجله مهندسی برق دانشگاه تبریز، جلد ۵۳، شماره ۳، صفحات ۲۲۳–۲۳۳، ۱۴۰۲.

[10] A. Kukkar, G. Kaur, “A novel adaptive ensemble classifier with LIME and SHAP-Based interpretability for fake news detection”, Expert Systems with Applications, 281, p.127751, 2025.

[11] D. Mehta, M. Patel, A. Dangi, N. Patwa, Z. Patel, R. Jain, P. Shah, B. Suthar, “Exploring the efficacy of natural language processing and supervised learning in the classification of fake news articles,” Advances of Robotic Technology, vol. 2, no. 1, 2024.

[۱۲] حسین عباسی مهر، محمد خودی‌زاده نهاری، "ارائه مدلی جهت پیش بینی ارتباط بین واحدهای دانشی در وب سایت های پرسش و پاسخ برنامه نویسی با استفاده از تکنیک های یادگیری عمیق"، مجله مهندسی برق دانشگاه تبریز، جلد ۵۴، شماره ۱، صفحات ۴۵-۵۴، ۱۴۰۳.

[13] T.H. Vo, T.L.T. Phan, K.C. Ninh, “Development of a fake news detection tool for Vietnamese based on deep learning techniques,” Unpublished or missing journal details, 2022.

[14] T. Mahmud, I. Hasan, M.T. Aziz, T. Rahman, M.S. Hossain, K. Andersson, “Enhanced fake news detection through the fusion of deep learning and repeat vector representations,” in Proc. 2024 2nd Int. Conf. Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp. 654–660, 2024.

[15] K. Zaheer, M.R. Talib, M.K. Hanif, M.U. Sarwar, “A Multi-Kernel Optimized Convolutional Neural Network With Urdu Word Embedding to Detect Fake News,” IEEE Access, 2023.

[16] C. Xu, M. T. Kechadi, “An enhanced fake news detection system with fuzzy deep learning”, IEEE Access, vol. 12, pp. 88006–88021, 2024.

[17] J. Hamidzadeh, Z. Mehravaran, A. Harati, “Feature selection by utilizing kernel-based fuzzy rough set and entropy-based non-dominated sorting genetic algorithm in multi-label data”, Knowledge and Information Systems, vol. 67, no. 4, pp. 3789–3819, 2025.

[18] X. Song, G. Yue, N. Mac Parthaláin, Y. Qu, “Mixture kernel-based fuzzy-rough feature selection”, in Proc. UK Workshop on Computational Intelligence, Cham: Springer Nature Switzerland, pp. 3–12, Sept. 2022.

[19] Y. Singh, P. Singh, “Fake News Detection Using LSTM in TensorFlow and Deep Learning,” Journal of Applied Science and Education (JASE), vol. 3, no. 2, pp. 1–14, 2023.

[20] F.K. Alarfaj, J.A. Khan, “Deep Dive into Fake News Detection: Feature-Centric Classification with Ensemble and Deep Learning Methods,” Algorithms, vol. 16, no. 11, p. 507, 2023.

[21] M.F. Mridha, A.J. Keya, M.A. Hamid, M.M. Monowar, M.S. Rahman, “A comprehensive review on fake news detection with deep learning,” IEEE Access, vol. 9, pp. 156151–156170, 2021.

[22] J. Ali Reshi, R. Ali, “An efficient fake news detection system using contextualized embeddings and recurrent neural network,” Unpublished or missing journal details, 2024.

[23] A.R. Mahlous, A. Al-Laith, “Fake News Detection in Arabic Tweets during the COVID-19 Pandemic,” Int. J. Adv. Comput. Sci. Appl, vol. 12, no. 6, pp. 778–788, 2021.

[24] W.A.F.B. Nordin, R. Alfred, C.P. Yee, S.H. Tanalol, R.V. Loudin, and Z. Iswandono, “Malay fake news classification using a deep learning approach,” in Proc. Int. Conf. Computational Science and Technology, Singapore: Springer Nature Singapore, Aug. 2022, pp. 17–32.

[25] D. G. Dev, V. Bhatnagar, “Hybrid RFSVM: Hybridization of SVM and Random Forest Models for Detection of Fake News,” Algorithms, vol. 17, no. 10, p. 459, 2024.

[26] R. Dutta, D.R. Das Adhikary, and M. Majumder, “A Deep Learning Model for Classification of COVID-19 Related Fake News,” in Electronic Systems and Intelligent Computing: Proceedings of ESIC, Singapore: Springer Nature Singapore, pp. 449–456, 2022.

[27] P. Rani, J. Shokeen, “FNNet: a secure ensemble-based approach for fake news detection using blockchain,” The Journal of Supercomputing, pp. 1-38, 2024.

[28] W. Xu, J. Wu, Q. Liu, S. Wu, L. Wang, “Evidence-aware fake news detection with graph neural networks”, In Proceedings of the ACM Web Conference, April 2022, pp. 2501–2510.

[29] A. H. J. Almarashy, M. R. Feizi-Derakhshi, P. Salehpour, “Elevating fake news detection through deep neural networks, encoding fused multi-modal features,” IEEE Access, 2024.

[30] I. Ahmad, M. Yousaf, S. Yousaf, M. O. Ahmad, “Fake news detection using machine learning ensemble methods,” Complexity, vol. 2020, no. 1, p. 8885861, 2020.

[31] M. Luqman, M. Faheem, W. Y. Ramay, M. K. Saeed, M. B. Ahmad, “Utilizing ensemble learning for detecting multi-modal fake news,” IEEE Access, 2024.

[32] A. H. E. Wynne, K. T. Swe, “A Soft Two-Layers Voting Model for Fake News Detection,” 2024.

[33] S. V. Balshetwar, A. Rs, “Fake news detection in social media based on sentiment analysis using classifier techniques,” Multimedia Tools and Applications, vol. 82, no. 23, pp. 35781-35811, 2023.

[34] L.D.F. Santos, M.V. da Silva, “The effect of stemming and lemmatization on Portuguese fake news text classification,” arXiv preprint arXiv, vol. 2310, p.11344, 2023.

[35] A. Kishwar, A. Zafar, “Fake news detection on Pakistani news using machine learning and deep learning,” Expert Systems with Applications, vol. 211, p. 118558, 2023.

[36] H. Saleh, A. Alharbi, S.H. Alsamhi, “OPCNN-FAKE: Optimized convolutional neural network for fake news detection,” IEEE Access, vol. 9, pp. 129471-129489, 2021.

[37] D. Khyani, B.S. Siddhartha, N.M. Niveditha, B.M. Divya, “An interpretation of lemmatization and stemming in natural language processing,” Journal of University of Shanghai for Science and Technology, vol. 22, no. 10, pp. 350-357, 2021.

[38] J. Asha, A. Meenakowshalya, “Fake news detection using n-gram analysis and machine learning algorithms,” Journal of Mobile Computing, Communications & Mobile Networks, vol. 8, no. 1, pp. 33-43, 2021.

[39] A. Mallik, S. Kumar, “Word2Vec and LSTM based deep learning technique for context-free fake news detection,” Multimedia Tools and Applications, vol. 83, no. 1, pp. 919-940, 2024.

[40] B. Jang, M. Kim, G. Harerimana, S.U. Kang, J.W. Kim, “Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism,” Applied Sciences, vol. 10, no. 17, p. 5841, 2020.

[41] U. Sharma, S. Saran, S.M. Patil, “Fake news detection using machine learning algorithms,” International Journal of Creative Research Thoughts (IJCRT), vol. 8, no. 6, pp. 509–518, 2020.

[42] K. A. Kamel, J. Waleed, “The detection of fake text news using a dense-based 1D-CNN deep learning algorithm,” Academic Science Journal, vol. 2, no. 2, pp. 156–171, 2024.

[43] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, “Fake news detection on social media: A data mining perspective”, ACM SIGKDD Explorations, vol. 19, no. 1, pp. 22–36, 2017.

[44] A. Bondielli, F. Marcelloni, “A survey on fake news and rumor detection techniques”, Information Sciences, vol. 497, pp. 38–55, 2019.