Received 15 November 2023, accepted 29 November 2023, date of publication 5 December 2023,
date of current version 15 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339621

RESEARCH ARTICLE

Enhancing Fake News Detection by Multi-Feature Classification

AHMED HASHIM JAWAD ALMARASHY[1], MOHAMMAD-REZA FEIZI-DERAKHSHI[1], AND PEDRAM SALEHPOUR[2], (Member, IEEE)
[1]ComInSys Laboratory, Department of Computer Engineering, University of Tabriz, Tabriz 51666, Iran
[2]Department of Computer Engineering, University of Tabriz, Tabriz 51666, Iran

Corresponding author: Mohammad-Reza Feizi-Derakhshi (mfeizi@tabrizu.ac.ir)

ABSTRACT The proliferation of social media platforms has significantly accelerated our access to news, but it has also facilitated the rapid dissemination of fake news. Automatic fake news detection systems can help solve this problem. Although there is much research in this area, getting an accurate detection system is still a challenge. This article proposes a novel model to increase the accuracy of fake news detection. The theory behind the proposed model is to extract and combine global, spatial, and temporal features of text to use in a new fast classifier. The proposed model consists of two phases: first, global features are extracted by TF-IDF, spatial features by a convolutional neural network (CNN), and temporal features by bi-directional long short-term memory (BiLSTM) simultaneously. Then a fast learning network (FLN) is used to efficiently classify the features. Extensive experiments were conducted using two publicly available fake news datasets: ISOT and FA-KES. These two have different sizes; therefore, the proposed architecture (CNN+BiLSTM+FLN) can be evaluated much better. Results demonstrate the proposed model's superiority in comparison with previous works.

INDEX TERMS Fake news, social media platforms, distinguishing real and fake news, global, temporal, spatial features, novel architecture: CNN+BiLSTM+FLN, convolutional neural network (CNN), bi-directional long short-term memory (BiLSTM), fast learning network (FLN).

I. INTRODUCTION

The exponential growth of online news platforms, including social media, digital news sources, and traditional print media, has facilitated the rapid dissemination of fake news. This issue arises from the ease of uploading content onto these platforms, leading to a significant portion of the global population relying on social media channels such as Twitter, Facebook, Instagram, and YouTube as their primary source of news and information. This reliance is particularly prominent in developing nations where access to traditional news outlets is limited. Consequently, individuals across different geographical locations exploit these ubiquitous social media platforms to spread false information through various networking channels, often with illicit intentions. The increasing prevalence of social media usage has profound impacts on society, commerce, and culture, yielding both advantageous and disadvantageous outcomes. One of the most significant risks associated with this phenomenon is the dissemination of false information, which can have far-reaching consequences, including undermining worldwide trade, journalism, and democratic processes. The issue of fake news gained significant attention in 2016 following the former presidential election in the United States [1]. For instance, a fabricated news story in 2013, falsely claiming that President Barack Obama had been injured in an explosion, resulted in a stock market deficit of 130 billion dollars [2]. Scholars at Stanford University have provided statistical data indicating that a substantial proportion of fake news, amounting to 72.3 percent, can be attributed to both traditional news sources and digital social media platforms. The detrimental impact of misinformation on the general populace is widely acknowledged and poses a formidable obstacle to global trade, journalism, and governance. Given the peril posed by the proliferation of fabricated information, it becomes imperative to develop effective mechanisms

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Guidi.

for identifying fake news. Fake news encompasses various categories such as misinformation, disinformation, satire or parody, clickbait, conspiracy theories, political propaganda, and hoaxes, each with its characteristics and intent. These categories may overlap, and sometimes it can be challenging to precisely classify fake news into a single category due to the complexity of the content. The impact of different categories of fake news on the performance of the detection system is an essential aspect that warrants thorough investigation. Addressing this concern will provide a more nuanced understanding of the detection system's strengths and limitations as categorization of fake news, performance evaluation by category, comparison of categories, and analyze the practical implications of the performance variations across different categories [3].

The field of computer science is witnessing an upsurge in experimentation as Artificial Intelligence progresses rapidly. Researchers are now addressing the novel challenge of detecting fake news, which has not been previously tackled. Machine Learning (ML)-based automatic detection approaches have been extensively studied to combat the spread of fake news. These systems utilize ML techniques to aid consumers in evaluating the veracity of the content they encounter, enabling them to determine whether a given news piece is genuine or not. Recent advancements in Deep Learning (DL) techniques have further improved the effectiveness and efficiency of fake news detection, surpassing the capabilities of traditional ML methods.

The fake news detection process can be considered a classification system in artificial intelligence in which a classifier tries to classify input text into "fake" or "real" news. Previous research has aimed to create accurate classification systems, but achieving optimal performance remains an ongoing pursuit. This article presents a research focus on developing a reliable and precise automated system for detecting fake news on social media.

In the proposed deep neural network architecture, the convolutional neural network (CNN) and bi-directional long short-term memory (BiLSTM) are employed to simultaneously extract spatial and temporal features. Also, the TF-IDF (Term Frequency-Inverse Document Frequency) technique is used to capture global features. These features are then combined and classified using a Fast Learning Network (FLN) [4]. The FLN algorithm, which is known for its dual parallel forward neural network configuration, demonstrates superior regression accuracy, generalization performance, stability, and quick convergence.

This approach ensures the extraction of a diverse range of features, making it highly effective in detecting fake news. Two famous and publicly available fake news datasets, ISOT and FA-KES, are used in this study.

Both the title and body text of these two datasets are used as input in this system. Various evaluation metrics, including accuracy, recall, F_1, and precision, are employed to assess the performance of the proposed method. The experimental results are presented using both tabular and graphical formats, aligning with the latest research findings for better comprehension.

The contributions of this paper are as follows: Integration of CNN+BiLSTM and TF-IDF within the deep neural network architecture. The theory behind this integration is that CNN deals with spatial features, BiLSTM is well-known for extracting temporal features, and the traditional TF-IDF technique can extract global features of text. Using these three components at the same time gives a better view of the input text to the classifier, resulting in improved detection accuracy. The results of experiments confirm that the effectiveness of detection increases with the incorporation of various features. Using two different datasets of different sizes Testing the model on two different datasets, which have different sizes and features, helps us evaluate the method much better and shows its ability to deal with different datasets with different scales and formats. Utilization of the FLN algorithm. Using FLN for the classification phase to leverage the effectiveness of deep neural networks. The FLN algorithm is well-known for its superior regression accuracy, generalization performance, stability, and quick convergence. The results of experiments confirm that the FLN algorithm outperforms comparable methods in terms of regression accuracy, generalization performance, and stability while maintaining rapid convergence. The subsequent sections of this work are organized as follows: Section II provides a comprehensive review of the relevant literature. Section III discusses the datasets and methodologies employed in this research. The experimental findings are presented in Section IV. Finally, Section V concludes the paper and provides recommendations for future work. The study utilizes two datasets: ISOT and FA-KES datasets to facilitate the research analysis.

II. LITERATURE REVIEW

The surge in deceptive content on social media has prompted researchers to intensify their efforts to find solutions. Several studies have been conducted in this field, and we highlight a selection of them below. Earlier studies in this field, like many other fields of NLP, were focused on probabilistic methods like SVM. However, after introducing deep learning, many researchers moved forward with DL. First, studies focused on comparing probabilistic methods and DL methods. Then they moved towards proposing more complicated DL models. For example, Article [5] provides a framework for selecting between DL and ML methods for problem-solving. It examines the accuracy of techniques like Naive Bayes and clustering and compares them against traditional methods. The study analyses the advantages and disadvantages of various DL techniques and their performance compared to conventional methods. In [6], a paradigm and procedure for identifying fake news are presented. The authors employ ML and NLP to gather news and utilize support vector machines to determine the veracity of the news. Reference [9] presents a system for identifying fake news based on feature extraction, feature selection, and vote classifiers. The

TABLE 1. Survey of several recent fake news detection publications.

Ref.	Year	Contributions	Datasets	Proposed Model
[27]	2023	The model adopted forward and backward of texts for fake news detection.	ISOT, FA-KES	(DeepCnnLstm + DeepCnnBilstm)
[26]	2023	The model adopted placements, and hyperparameter tuning for an effective solution for fake news detection.	ISOT, FA-KES	HyproBert model
[25]	2022	The model was trained using a dataset that had been augmented.	BanFakeNews	AugFake-BERT model
[18]	2022	The model investigates the impact of topic labels for the fake news and introduce contextual information of news.	LIAR	FDML model
[7]	2021	Automatic detection methods based on machine learning and artificial intelligence	ISOT, FA-KES	CNN+RNN
[22]	2021	Proposed a novel stacking model with McNemar's test	ISOT, KDnugget	(CNN, LSTM, GRU)+(RF1, KNN, LR, SVM)
[8]	2021	The present research employs classifiers benefited from ML and DL methodologies to examine the conduct of various characteristics linked to Facebook profiles.	More than 15000 news contents from different Facebook users	LSTM, KNN, LR, SVM
[9]	2021	Three experiments were conducted to evaluate the effectiveness of feature extraction and selection techniques, namely TF-IDF with chi-square, TF-IDF with ANOVA, and DOC2VEC.	Fake-or-Real-News, Media-Eval, and ISOT	Chi-square +(NB, SVM, PA, RF, SGD, LR)+ Voting classifier
[10]	2021	The current research suggests developing an optimized Convolutional Neural Network (CNN) model, denoted as OPCNN-FAKE, to detect fake news.	Dataset1, FakeNewsNet, FA-KES5, ISOT	OPCNN-FAKE, LR, DT, RNN, LSTM, SVM
[11]	2021	A deep learning approach named FakeBERT, based on Bidirectional Encoder Representations from Transformers (BERT), has been proposed.	FAKE-NEWS	CNN+LSTM, BERT
[24]	2021	Presented a deep learning framework (BerConvoNet)	Dataset1, Dataset2, Dataset3, FakeNewsNet	BERT, CNN
[23]	2020	A linguistic model is proposed	FakeNewsNet	SVM, LR, NB, DT, RNN, LSTM
[12]	2020	An attention mechanism is proposed for the CNN+BiLSTM ensemble model.	1356 tweets and various media reports	CNN+BiLSTM
[13]	2020	Improve the accuracy of existing fake news detection using a Deep Convolutional Neural Network (FNDNet)	Kaggle fake news dataset	GloVe + CNN
[21]	2020	Using principal component analysis and the Chi-square test, a novel model is proposed.	Fake News Challenges (FNC) website	CNN-LSTM, PCA, Chi-square
[6]	2019	Combined ML+DL	Social media platforms	NB, SVM, NLP
[14]	2019	This study proposes a framework for extracting hybrid features from online news textual metadata.	ISOT, LIAR, FA-KES	TFIDF, SVM, DT, LR
[15]	2019	Address the stance detection problem proposed in the context of fake news challenge(FNC-1)	FNC-1 dataset	NLI, BiLSTM
[20]	2019	By employing transfer learning on a RoBERT, a deep bidirectional transformer language model, a large-scale language model was created for stance identification.	FNC-1 dataset	The RoBERTa (Robustly Optimized BERT Approach)
[17]	2018	Stacked Bi-LSTM layers	FNC-1 dataset	BiLSTM, CNN
[19]	2018	Extension of the general architecture based on a matrix of similarity	FNC-1 dataset	LSTM, CNN, sMemNN

proposed system distinguishes between fake and real news and outperforms previous works in terms of accuracy for the ISOT dataset. Deep learning models gain better performance in comparison with probabilistic models. The authors of [8] propose a method for automatically detecting fake news on Facebook using deep learning. They utilize Facebook account-related features and news content attributes to assess account behavior.

The suggested approach surpasses current techniques in terms of accuracy when evaluated on real-world data. To combat the issue of fake news, [10] suggests new methods based on machine learning and deep learning. The authors propose an enhanced convolutional neural network architecture called OPCNN-FAKE, which achieves exceptional performance across different datasets and outperforms other models in detecting fake news. Research in [7] suggests a novel deep learning architecture that combines recurrent and convolutional neural networks for the automated detection of fake news using machine learning and artificial intelligence. The model exhibits superior diagnostic results compared to non-hybrid baseline techniques and demonstrates promising generalizability across different datasets. In [11], a deep learning approach called FakeBERT is introduced, which addresses the challenge of ambiguity in natural language processing. By combining BERT with deep learning techniques, the proposed model achieves a high accuracy of 98.90% and outperforms existing models.

Other studies, such as [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], and [25], also present various approaches and architectures for fake news detection, showcasing their performance on different datasets and achieving high levels of accuracy. Despite the promising results demonstrated by current deep learning algorithms in fake news detection, there is still room for improvement in this field.

One of the most up-to-date models is presented in [26]. In this paper, a state-of-the-art model adopted placements and hyperparameter tuning for an effective solution for fake news detection named HyproBert, which was evaluated using two fake news datasets (ISOT and FA-KES) and achieved higher performance compared to other baseline and state-of-the-art models. Also, [27] proposed (DeepCnnLstm + DeepCnnBilstm) that achieved the best level of accuracy in the detection of fake news on FA-KES dataset. We have selected these recent papers to compare them with our method. Despite the promising results demonstrated by current deep learning algorithms in fake news detection, there is still room for improvement in this field. Table1 provides an overview of state-of-the-art models, their contributions, and the datasets used in the related works, offering further insights into fake news detection.

III. MATERIALS AND METHODS

The process of the proposed architecture comprises two primary phases: the initial phase involves the extraction of major features that are deemed to have a significant impact on the performance of the classifier. The subsequent stage involves the classification phase, wherein the features are categorized into two distinct classes, one denoting real and the other fake, as illustrated in Figure 1.

A. DATASET

The proposed study utilized two distinct datasets: ISOT [34] and FA-KES [35]. These datasets are without missing or outlier values.

ISOT dataset consists of real and fake news articles obtained from authentic real-world sources. The authors crawled articles from Reuters.com, a reputable news source, to collect real news articles, while fake news articles were gathered from websites that were deemed unreliable and had been flagged by Politifact and Wikipedia. These articles were primarily published between 2016 and 2017. It contains a total of 44,898 instances. It is balanced, with 21,417 instances labeled as real news (labeled as 1) and 23,481 instances labeled as fake news (labeled as 0). Sample of the ISOT can be found in Table 2.

FA-KES dataset focuses on news articles related to the Syrian war. It consists of 804 news articles, each including the headline, date, location, news sources, and the complete article body. The label for the class is set to '0' for fake news and '1' for real news. It is balanced, with 53% of the articles labeled as true and 47% labeled as fake. Sample of the FA-KES dataset can be found in Table 3.

TABLE 2. Presents a sample of the ISOT dataset.

Label	Title	Text
0 (fake news)	Drunk Bragging D. Trump Staffer Started Russian	House Intelligence C. Chairman Devin ....
0 (fake news)	White House Panics Knowing Flynn Is Going To Take Them Down	While Donald Trump has been taking vacations....
1 (real news)	Factbox: Provisions of the U.S. Rep. final tax bill	(Reuters) - Republicans U.S. reached ....
1 (real news)	Mulvaney says U.S. tax bill votes could be Wednesday: CNBC	(Reuters) – W. H. budget D. said ....
0 (fake news)	Bannon Takes His 'Weapons' At Breitbart; Smears 'Quran Kissing'	This week s chaotic news cycle was mostly dominated ....

TABLE 3. Presents a sample of the FA-KES dataset.

Label	Title	Text
1 (real news)	US-led air strikes in Syria kill dozens of civilians	Dozens of civilians have reportedly been killed by ....
0 (fake news)	Syrias Nusra Front stages deadly suicide bombing in Aleppo	A suicide bomber from al-Qaedas Syrian affiliate ....
0 (fake news)	Regime troops thwart rebel attack in Syrias Aleppo	Regime forces fended off an attack by rebel fighters on....
1 (real news)	String of bomb blasts kill at least 42 in Syria	At least 42 people were killed and dozens more wounded..
1 (real news)	Syrian Coalition: Assad Used Chemical Weapons 136 Times since	The Syrian Coalition called upon the UN Security ....

B. PRE-PROCESSING

The textual data undergoes several preprocessing steps, the text data is cleaned, standardized, and transformed into a suitable format for subsequent feature extraction and analysis.

The following steps are included in the preprocessing phase: Lowercasing: All text is converted to lowercase to ensure consistency and avoid treating words with different cases as distinct. Cleaning up: Various cleaning operations are performed to remove unwanted elements from the text. This includes deleting URLs, punctuation marks (including the hash character #), and special characters specific to platforms like Twitter ($, &, %, etc.). Non-ASCII English characters are also removed to preserve data exclusive to the English language. Replacing: Certain textual elements are replaced with their related words or simplified forms. This includes replacing contractions with their extended words (e.g., replacing "I'll" with "I will"), converting emoji to their corresponding words, and reducing repetitive occurrences of a character to a single occurrence (e.g., converting

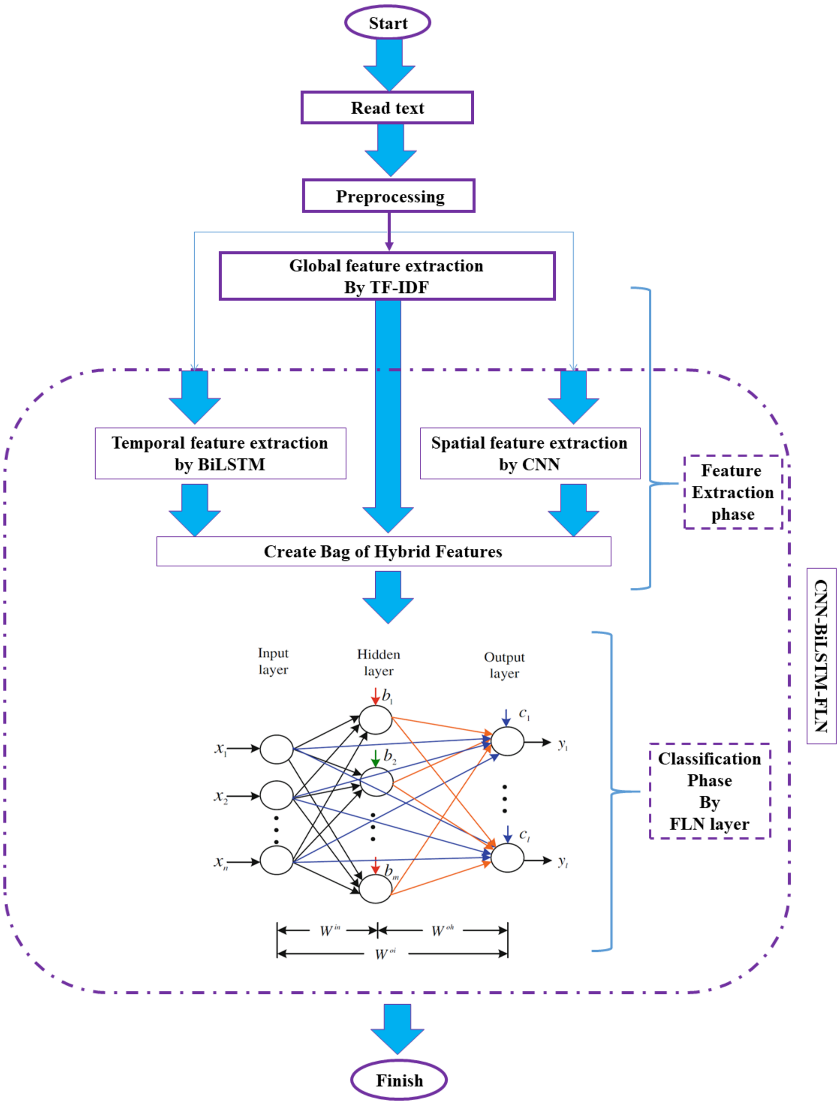

FIGURE 1. Overall proposed method architecture.

'happpppppy' to 'happy'). Tokenization: The cleaned text is then split into individual tokens or words to create a tokenized representation of the text. This step helps with further processing and analysis. Lemmatization/Stemming: Each token is further processed by applying lemmatization or stemming techniques. Lemmatization aims to convert the tokens to their base or root form, such as converting the token "interesting" to "interest." Stemming, on the other hand, involves reducing the tokens to their stem form by removing prefixes or suffixes.

C. GLOBAL FEATURES EXTRACTION PHASE

The TF-IDF (Term Frequency-Inverse Document Frequency) method is used to extract global features [28]. Here is an overview of the TF-IDF process:

Pre-processing: The text documents undergo pre-processing steps, including cleaning, lowercasing, removing special characters, and tokenization on the sentence level.

Term Frequency (TF) computation: TF represents the frequency of a term (word) in a document. It is the ratio of the number of times a specific word appears in a document to the total number of words in that document.

Document-Term Matrix: A table is generated to display the frequency of each word in each sentence. This matrix captures the occurrence of words in the documents. Inverse Document Frequency (IDF) computation: IDF represents the significance of a term across the entire document collection. It is the logarithm of the total number of documents divided by the number of documents that contain the specific word.

TF-IDF computation: The TF-IDF score for each word in a document is obtained by multiplying the TF value and IDF value as presented in Figure 2. This step assigns higher scores to words that are frequent in a document but relatively rare in the entire document collection. Threshold determination: The average score of all words is computed, and a threshold value

$$\boldsymbol{TF(t,d)} = \frac{(Number\ of\ occurrences\ of\ term\ t\ in\ document\ d)}{(Total\ number\ of\ terms\ in\ the\ document\ d)}$$

$$\boldsymbol{IDF(t,D)} = \log_e \frac{(Total\ number\ of\ documents\ in\ the\ corpus)}{(Number\ of\ documents\ with\ term\ t\ in\ them)}$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

FIGURE 2. **TF-IDF(term frequency-inverse document frequency).**

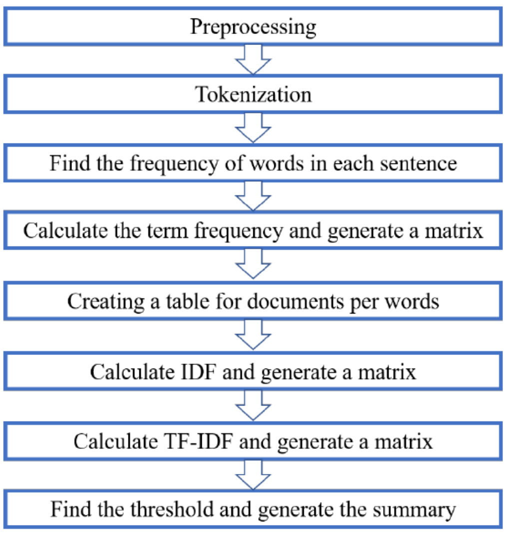

FIGURE 3. **Global features extraction process.**

is determined. Words with scores higher than the threshold value are considered eligible for selection.

Global feature extraction: The TF-IDF process extracts global features by considering the significance of words across the entire dataset, rather than just within individual documents. The overall procedure of the TF-IDF process is summarized in Figure 3, which provides an illustration of the steps involved in converting text documents into vector representations using TF-IDF.

D. LOCAL AND TEMPORAL FEATURES EXTRACTION PHASE

In the feature extraction phase, the study leverages the power of deep neural networks, particularly a hybrid model called CNN+BiLSTM [29], to capture both spatial and sequential features from the text data. The CNN+BiLSTM architecture consists of two main components:

Convolutional Neural Network (CNN): A one-dimensional CNN (Conv1D) is utilized to process the input vectors, which represent the pre-processed and transformed text data. The Conv1D layer is capable of extracting local features by applying filters over the input data and capturing patterns that are relevant to the task at hand. In this case, the CNN focuses on capturing spatial features from the text.

Bidirectional Long Short-Term Memory (BiLSTM): A BiLSTM layer is employed to capture temporal features by modeling the sequential relationships between words and phrases in both forward and backward directions. The BiLSTM layer is a type of recurrent neural network that is well-suited for processing sequential data. It is able to remember and propagate information over long distances, making it effective in capturing contextual information and dependencies in text.

By combining the CNN and BiLSTM layers in the architecture, the model is able to extract both local (spatial) and temporal features simultaneously. The CNN focuses on capturing patterns and features within short windows of the input, while the BiLSTM captures the sequential relationships and context across the entire text. This hybrid approach allows the model to capture a comprehensive range of features from the text data, enabling more effective and accurate analysis and classification tasks.

1) CNN CONTRIBUTION

We've provided an excellent overview of the advantages of using CNNs for feature extraction in Natural Language Processing (NLP) tasks. CNNs, although widely recognized for their success in computer vision, have also demonstrated their utility in NLP applications, including text categorization and sentiment analysis. One of the main advantages of CNNs in NLP is their ability to capture local patterns and dependencies within the text. By treating words or characters as one-dimensional signals, CNNs can apply one-dimensional convolutions (Conv1D) to extract local features. This is particularly useful for tasks like sentiment analysis or identifying key features in text classification, where capturing n-grams or short phrases is important. CNNs are also flexible in handling variable-length inputs, making them suitable for processing documents of different lengths. This flexibility allows them to be applied to a wide range of NLP tasks without requiring fixed input sizes. The process of using CNNs for feature extraction in NLP typically involves the following steps:

Application of one-dimensional convolutions to the input representations, where filters slide over the input and perform element-wise multiplications to create feature maps. Multiple filters can be used to capture different types of features.

Utilization of pooling layers, such as max pooling, to down-sample the feature maps and reduce their dimensionality while retaining important information. Pooling helps capture significant features and introduces a level of interpretation independence.

Application of activation functions, such as ReLU, element-wise to introduce non-linearity into the network, enabling the learning of complex patterns. Training the CNN involves specifying the number of filters and the kernel size. Conv1D is commonly used in text categorization and NLP tasks, as it is designed to process one-dimensional sequences of word vectors. Figure 4 depicts the procedure of Conv1D graphically.

By stacking multiple convolutional layers with different filter sizes and hyper parameters, CNNs can learn hierarchical

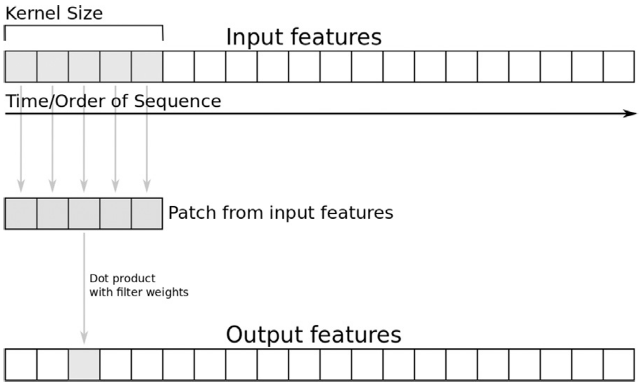

FIGURE 4. A 1-D convolutional operation.

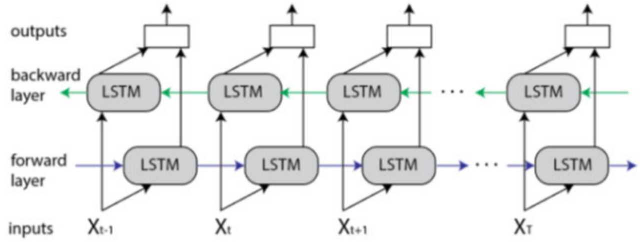

FIGURE 5. BiLSTM network.

representations of the text and capture local features at different levels. This allows them to extract meaningful information from the input data.

Furthermore, CNNs can be combined with other architectures, such as recurrent or transformer-based models, to leverage their respective strengths.

For example, in a hierarchical CNN, lower layers may capture word-level features, while higher layers capture sentence-level features.

While CNNs may not capture long-term dependencies as effectively as recurrent or transformer models, they still offer value in feature extraction for specific NLP tasks, especially when local patterns and relationships are critical. The design and architecture of the CNN can be tailored to meet the requirements of the task at hand.

2) BILSTM CONTRIBUTION

We have provided an accurate description of the Bidirectional Long Short-Term Memory (BiLSTM) networks and their significance in Natural Language Processing (NLP) tasks for feature extraction.

BiLSTMs are a variant of recurrent neural networks (RNNs) that are widely used in NLP due to their ability to capture both past and future context of the input sequence. They process the input sequence in both forward and backward directions simultaneously, allowing them to capture dependencies and context from both directions.

In contrast to standard LSTMs, which only process the input sequence in a forward manner, BiLSTMs incorporate an additional LSTM layer that operates in reverse, enabling the flow of information from the end of the sequence to the beginning. This bidirectional flow of information enables the model to capture contextual information from both past and future dependencies, leading to a richer representation of the input sequence.

The primary role of BiLSTMs in NLP feature extraction is to capture bidirectional contextual information and encode it into feature representations. By considering both the past and future context, BiLSTMs can capture important patterns and relationships within the text, making them suitable for various NLP tasks such as sentiment analysis, machine translation, named entity recognition, and classification.

In a BiLSTM network, the input sequence is processed by two LSTM layers, one in the forward direction and the other in the backward direction. The outputs of both LSTM layers can be combined using different methods such as averaging, summing, multiplying, or concatenating to create a unified representation that captures both forward and backward context.

Figure 5 illustrates the structure of a BiLSTM network, showcasing the forward and backward LSTM layers and their combination to generate the final representation of the input sequence.

Overall, BiLSTMs are powerful tools for NLP feature extraction as they leverage bidirectional context to capture dependencies and contextual information from both past and future perspectives, leading to improved representation learning in various NLP tasks. BiLSTMs are well-suited for NLP tasks that involve processing input sequences where the order of words or characters is important. They are designed to handle sequential data and can efficiently process input sequences of varying lengths. BiLSTMs capture contextual information by considering both the past and future context of each word or character in the input sequence. The forward pass captures past context, while the backward pass captures future context, enabling the model to have a comprehensive understanding of the input. Unlike traditional LSTMs that can only capture dependencies in the past, BiLSTMs address this limitation by incorporating information from both directions [30].

This makes them effective in capturing long-term dependencies in the input sequence, which is crucial for various NLP tasks such as sentiment analysis, machine translation, and named entity recognition. The hidden states of the BiLSTM at each time step serve as rich feature representations of the input sequence. These hidden states encode contextual information and capture important patterns and relationships within the text. These representations can be used as inputs for subsequent layers or fed into classification or regression models for different NLP tasks.

BiLSTMs are often used to generate contextual word embeddings, such as ELMo, which enrich word representations with contextual information from surrounding words. This helps in capturing multiple meanings (polysemy) and resolving ambiguities in language understanding tasks.

BiLSTMs trained on large-scale language modeling tasks can be used as powerful feature extractors in transfer learning settings.

The lower layers of a pre-trained BiLSTM can be used as feature extractors, while the higher layers can be fine-tuned or replaced to adapt to specific downstream tasks.

Overall, BiLSTMs offer significant benefits in NLP feature extraction by capturing both past and future context, handling sequential data, and providing rich feature representations that can be utilized for a variety of NLP tasks.

E. FUSION FEATURES

In the study, a comprehensive feature set was created by combining features extracted from different sources using early fusion.

Early fusion refers to the process of combining features at an early stage of the model architecture [31]. The extracted features from different sources, such as TF-IDF representations, CNN-based local features, and BiLSTM-based temporal features, were fused together to create a unified representation for each input instance [32].

Early fusion can lead to better exploitation of synergies among data sources, potentially yielding improved accuracy and robustness [33]. However, it requires careful consideration of data preprocessing, feature selection, and fusion techniques to ensure optimal results. Additionally, early fusion may face challenges related to handling data heterogeneity, addressing feature misalignment, and dealing with missing or noisy data.

In summary, early fusion is a powerful technique for integrating information from multiple sources at an early stage, providing a comprehensive and enriched representation that can lead to improved performance in various applications. By performing early fusion, the model can leverage the combined information from multiple feature sources to make predictions or perform downstream tasks. This approach aims to capture complementary information and potentially enhance the overall performance of the model.

The specific details of how the features were combined or fused depend on the architecture and design choices of the model. It could involve concatenating the feature vectors, applying element-wise operations, or employing other fusion techniques to merge the information from different feature sources.

Overall, early fusion of features extracted from different sources allows the model to leverage diverse information and create a more comprehensive representation of the input data, potentially improving the performance of the model on the given NLP task.

F. CLASSIFICATION PHASE

We used a neural network structure called Fast Learning Network [4], as shown in Figure 6, for the classification phase of our proposed method.

The FLN described in our statement has a unique characteristic in its weight initialization process and weighted connections.

Weight Initialization: The input weights and biases of the hidden layer in the FLN are randomly generated. Random

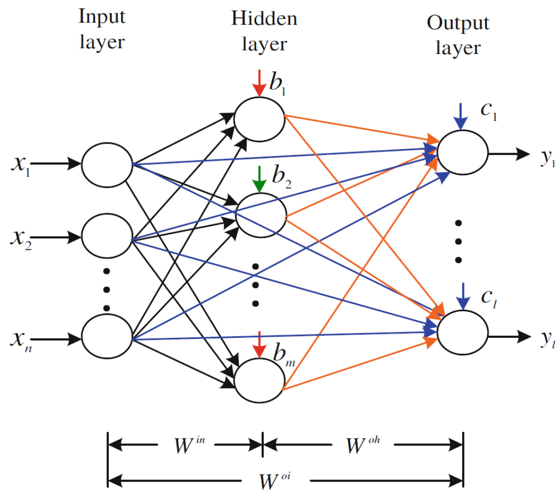

FIGURE 6. Structure of fast learning network [4].

weight initialization is common in neural networks to introduce diversity in the initial weights and avoid getting stuck in local optima during training. By starting with random weights, the network has a chance to explore different regions of the weight space and find better solutions.

Weighted Connections: The FLN has weighted connections between the output layer and the input layer, as well as between the output nodes and the hidden nodes.

These weighted connections determine the flow of information between the layers and nodes of the network. The weighted values of these connections are determined using least squares methods, which are analytical techniques used for finding the best-fit line or curve to a set of data points.

By connecting the input nodes not only to the hidden layer but also directly to the output nodes, the FLN aims to increase the learning speed and improve the accuracy of the network. This connectivity pattern allows information to flow more directly from the input to the output, potentially enabling the network to capture important features and make accurate predictions.

Suppose, there are N arbitrary distinct samples $\{x_i,y_i\}$, in which $x_i=[x_{i1}, x_{i2},\ldots x_{in}]^T \in R^n$ is the n-dimensional feather vector of the ith sample, and $y_l=[y_{l1}, y_{l2},\ldots y_{ln}]^T \in R^l$ is the corresponding l-dimensional output vector.

The FLN has m hidden layer nodes. W^{in} is the m $\times$ n input weight matrix, b=$[b_1,b_2,\ldots,b_m]$ is the biases of hidden layer nodes, and W^{oh} is a $l \times m$ matrix which consists of the weight values of the connection between the output layer and the hidden layer.

W^{oi} is a $l \times n$ weight matrix which contains weight values of the connection between the output layer and the input layer. c=$[c_1,c_2,\ldots,c_l]^T$ is the biases of output layer nodes. $g(.)$ and $f(.)$ are the active functions of hidden nodes and output nodes.

Then, the FLN is mathematically modeled as equation 1:

$$\mathbf{y}_j = f(W^{oi}x_j + c + \sum_{k=1}^{m} w_k^{oh} g(w_k^{in}x_j + b_k)) \tag{1}$$

where the parameters of equation 1 are:

j=1,2,…,N

W^{oi}: The weight vector connecting jth output nodes and input nodes, $W^{oi} = [W_1^{oi}, W_2^{oi}, \ldots, W_l^{oi}]$.

W_k^{oh}: The weight vector connecting the kth nodes of the hidden layer and the nodes of the output layer, $W_k^{oh} = [W_{k1}^{oh}, W_{k2}^{oh}, \ldots, W_{kl}^{oh}]^T$.

W_k^{in}: The weight vector connecting the kth nodes of the hidden layer and the nodes of the input layer,

$W_k^{in} = [W_{k1}^{in}, W_{k2}^{in}, \ldots, W_{km}^{in}]^T$.

Then equation 1 can be written compactly as equation 2.

$$\mathbf{Y} = f(W^{oi}X + W^{oh}G + c) = f(W \begin{bmatrix} X \\ G \\ I \end{bmatrix}) \tag{2}$$

Learning algorithm for FLN: Suppose we have a training set s including input x and target y, where the number of hidden layer nodes is m and the activation function of hidden nodes is g(.), the FLN algorithm can be defined as follows: The matrix of W^{in} input weights as well as the matrix of bias values b are randomly generated. The hidden layer output matrix of FLN (G) is calculated using equation 3.

$$\mathbf{G} = \begin{bmatrix} g(W_1^{in}x_1 + b_1) & \ldots & g(W_1^{in}x_N + b_1) \\ \vdots & \ddots & \vdots \\ g(W_m^{in}x_1 + b_m) & \ldots & g(W_m^{in}x_N + b_m) \end{bmatrix}_{m \times N} \tag{3}$$

$$\mathbf{W} = [W^{oi}W^{oh}c]_{(l \times (n+m+1))} \tag{4}$$

The matrix W = $[W^{oi}W^{oh}c]$ could be called as output weights.

$$\mathbf{I} = [11\hat{a} \ldots 1]_{1 \times N} \tag{5}$$

The minimum norm least-squares solution of the linear system could be written as equation 6.

$$\hat{W} = f^{-1}(Y) \begin{bmatrix} X \\ G \\ I \end{bmatrix}^+ = f^{-1}(Y)H^+ \tag{6}$$

where $H=[X^T G^T I^T]$, and also $f^{-1}(.)$ is the invertible function of the activation function f(.) of output nodes.

Finally, the determination of the weights of the FLN network is achieved through the utilization of the equations 7, 8, and 9:

$$W^{oi} = \hat{W}(1:l, 1:n) \tag{7}$$
$$W^{oh} = \hat{W}(1:l, n+1:(n+m)) \tag{8}$$
$$c = \hat{W}(1:l, n+m+1) \tag{9}$$

In addition, in FLN, the input weights and hidden layer biases are calculated haphazardly, whereas the other weights may be found analytically through the use of least squares methods. Therefore, the FLN was able to overcome the majority of the drawbacks presented by traditional ways of learning while also possessing an extremely rapid learning speed.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The study began by cleansing the dataset, ensuring it was free from any irrelevant or noisy information. The applied tokenization technique then represents the text and captures its characteristics. The model simultaneously extracted global, local, and temporal features. These features were combined and classified by the FLN classifier.

A. EXPERIMENTAL AND HYPERPARAMETER SETTINGS

The datasets used in the study, namely ISOT and FA-KES, were divided into training and testing sets. 80% of the instances from both datasets were allocated for training, while the remaining 20% were used for testing.

For the implementation of our models, we worked on an HP Core i7 computer with 4 GB of RAM and a 64-bit operating system. The experiments were conducted using MATLAB R2022a software. The learner algorithm for CNN is the stochastic gradient descent algorithm; it evaluates the gradient and updates the parameters using a subset of the training data. A different subset, called a mini-batch, is used at each iteration. The full pass of the training algorithm over the entire training set using mini-batches is one epoch. Stochastic gradient descent is stochastic because the parameter updates computed using a mini-batch are a noisy estimate of the parameter update that would result from using the full data set.

The learner algorithm for CNN is the stochastic gradient descent algorithm; it evaluates the gradient and updates the parameters using a subset of the training data. A different subset, called a mini-batch, is used at each iteration. The full pass of the training algorithm over the entire training set using mini-batches is one epoch. Stochastic gradient descent is stochastic because the parameter updates computed using a mini-batch are a noisy estimate of the parameter update that would result from using the full data set.

In BiLSTM, the Adam optimization algorithm is used to adapt the learning rate of each parameter to achieve better and faster convergence of the training process. Table 4 shows the network's hyperparameters.

The proposed deep network underwent a learning process on both the ISOT and FA-KES databases. For the ISOT dataset, the training process consisted of 1000 iterations, while for the FA-KES dataset, it was performed over 1200 iterations. After training, the network was evaluated using the test data.

The average time required to train the proposed network was 45 seconds for the ISOT dataset and 52 seconds for the FA-KES dataset. Figure IV-A depicts the classification accuracy in the evaluation mode and the training error for the ISOT dataset. It can be observed that the network converged to 99% accuracy after 100 iterations, and the training error remained below 0.01 within the same iterations.

TABLE 4. Network's hyperparameters.

Parameter	Value
CNN Learner Algorithm	sgdm
CNN Maximum Number of Epochs	200
CNN InitialLearnRate	0.1
CNN LearnRateDropFactor	0.2
CNN LearnRateDropPeriod	5
BiLSTM Learner Algorithm	adam
BiLSTM Maximum Number of Epochs	25
BiLSTM InitialLearnRate	0.1
BiLSTM LearnRateDropFactor	0.2
BiLSTM LearnRateDropPeriod	5
MiniBatchSize	128

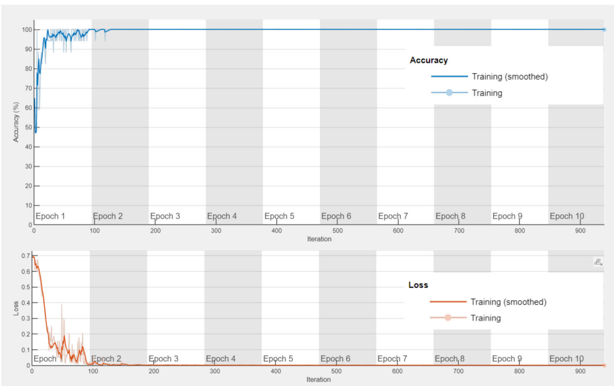

FIGURE 7. Training accuracy and loss for proposed network on ISOT dataset.

On the other hand, Figure 8 illustrates the convergence results and training error of the proposed network on the FA-KES database.

Although the network on the FA-KES database did not converge as well as on the ISOT database, it still exhibited excellent performance.

This fluctuation in convergence can be attributed to the limited number of training samples available in the FA-KES dataset. Overall, the proposed network demonstrated strong performance on both the ISOT and FA-KES databases, achieving high accuracy even with the challenges posed by smaller datasets.

B. MODEL EVALUATION CRITERIA

The evaluation of the model was conducted using accuracy (Acc), precision (Pr), recall (Re), and F_1 statistics.
According to Table 5, we compute the following metrics and mathematically expressed them in Equations 10, 11, 12, and 13.

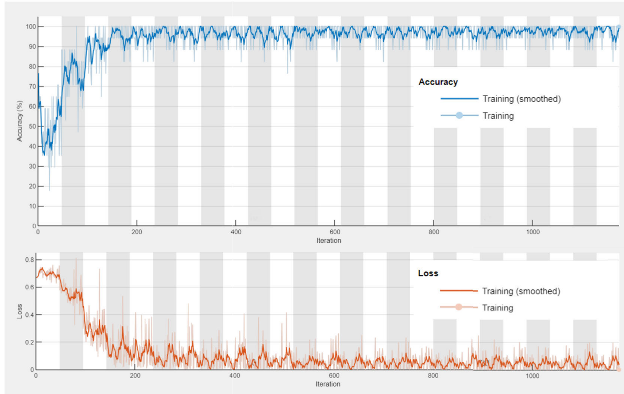

FIGURE 8. Training accuracy and loss for proposed network on FA-KES dataset.

TABLE 5. Confusion matrix.

Actual	Predicted:Real news	Predicted:Fake news
Real news	TP	FN
Fake news	FP	TN

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Accuracy refers to the proportion of correct predictions.

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

Recall refers to the classifier's capacity to identify all positive samples.

$$Precesion = \frac{TN}{TN + FP} \tag{12}$$

Precision measures the accuracy of positive predictions.

$$F_1 = 2\frac{RePr}{Re + Pr} \tag{13}$$

The F_1 is the harmonic mean of precision and recall, which computes values between 0 and 1.

C. RESULTS AND DISCUSSION ON THE ISOT DATASET

Table 6 and Figure IV-C present the results of various methods applied to the ISOT dataset, including the proposed method as well as previous methods discussed in the literature review. The comparison reveals that our proposed method performs better than other methods in terms of accuracy.

All models trained on the ISOT dataset demonstrate exceptional performance in detecting fake news. This can be attributed to the dataset's inclusion of long words, which provide more prominent features that contribute to the superior results achieved compared to the other datasets.

TABLE 6. Highest results of various methods on the ISOT.

Method	Acc	Re	Pr	F_1
RF [7]	92%	92%	92%	92%
KNNs [7]	60%	61%	67%	56%
DT [7]	96%	96%	96%	96%
CNN [7]	99%	99%	99%	99%
RNN [7]	98%	98%	98%	98%
Hybrid CNN+RNN [7]	99%	99%	99%	99%
HyproBert [26]	99.3%	99%	99%	99%
DeepCnnBilstm [27]	99%	100%	99%	99%
Our proposed model	99.4%	99%	99%	99%

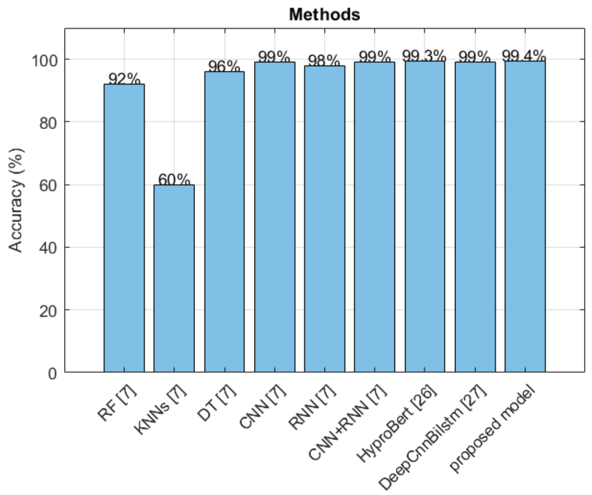

FIGURE 9. Comparison of different methods in term of accuracy for ISOT dataset.

It is worth noting that the performance of the proposed method for the ISOT database is comparable to the results reported in [7], [26], and [27] . This similarity in performance may be attributed to the large number of training samples available in the ISOT dataset.

The findings indicate that the proposed method exhibits a high level of performance on the ISOT dataset, achieving accuracy levels comparable to state-of-the-art approaches. This demonstrates the efficacy of the hybrid CNN+BiLSTM+FLN approach in accurately classifying fake news, even when dealing with datasets that contain a large number of training samples.

D. RESULTS AND DISCUSSION ON THE FA-KES DATASET

The results presented in Table 7 and Figure 10 indicate that the hybrid CNN+BiLSTM+FLN approach outperforms other methods in terms of F_1, accuracy, precision, and recall on the FA-KES dataset. Although modest in size, this database is used to evaluate the system's responsiveness and robustness with databases of various sizes. Despite the FA-KES dataset having fewer training samples compared to the ISOT dataset, the proposed method demonstrates superior performance by effectively combining CNN, BiLSTM, and FLN networks. This combination enables the learning process to be completed with a limited number of training samples while achieving high accuracy and speed in classification. The comparison of our proposed method with the approaches in [7], [26], and [27] demonstrates its superiority in terms of F_1, accuracy, precision, and recall, across a wide range of database sizes and formats. This is particularly noteworthy for small datasets where the proposed method excels. The provided Figures (IV-D and 12) display the comparison of the proposed method with other approaches in terms of precision, recall, and F_1 criteria for both the ISOT

TABLE 7. Highest results of various methods on the FA-KES.

Method	Acc	Re	Pr	F_1
RF [7]	53%	53%	56%	54%
KNNs [7]	57%	57%	58%	57%
DT [7]	55%	55%	56%	55%
CNN [7]	50%	50%	55%	48%
RNN [7]	50%	50%	51%	50%
Hybrid CNN+RNN [7]	60%	60%	59%	59%
HyproBert [26]	61%	61%	60%	61%
DeepCnnBilstm [27]	68%	88%	64%	74%
Our proposed model	99%	99%	99%	99%

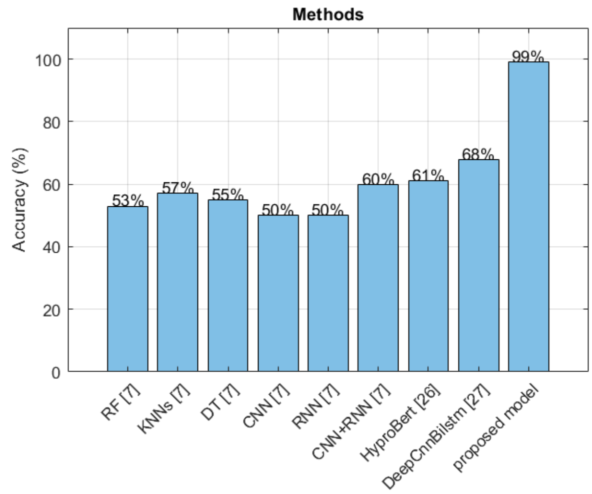

FIGURE 10. Comparison of different methods in term of accuracy for FA-KES dataset.

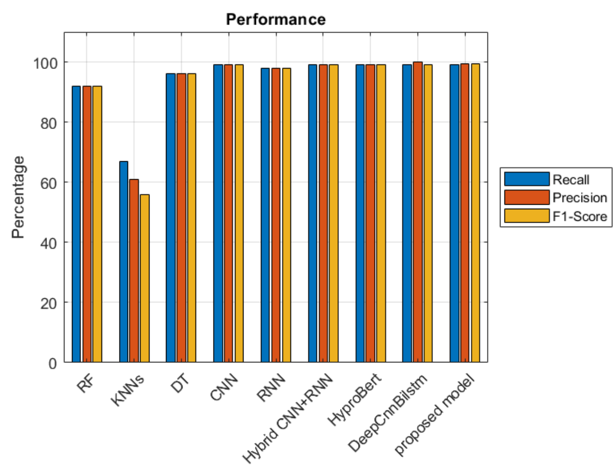

FIGURE 11. Comparison of the performance of our proposed method with other methods on the ISOT dataset.

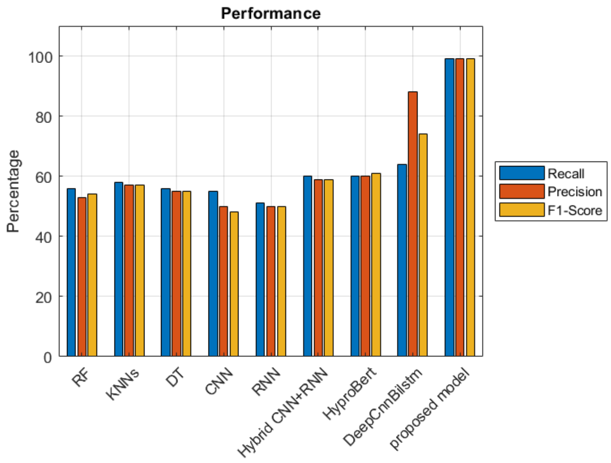

FIGURE 12. **Comparison of the performance of our proposed method with other methods on the FA-KES dataset.**

and FA-KES databases, further highlighting the superior performance of the hybrid CNN+BiLSTM+FLN approach. (We noticed that there are two typos in [7] in the F_1 measure but we just left it to keep the results of the original paper).

Overall, these results validate the effectiveness of the proposed method in achieving high accuracy and performance in fake news detection, particularly on datasets with limited training samples.

V. CONCLUSION AND FUTURE WORK

In this paper, we mentioned focusing on evaluating the effectiveness of a deep neural network architecture for fake news detection using two different datasets. The main contributions of the research include the adoption of deep neural networks and TF-IDF for feature extraction, testing the model on different-sized datasets, and utilizing the FLN during the classification phase. The theory behind deep neural networks for feature extraction lies in the ability of CNN to extract spatial features and the ability of BiLSTM to extract temporal features, and we used a parallel combination of them in our proposed method. The proposed approach achieved high levels of accuracy on both datasets, surpassing the performance of other comparable methods.

The experiments have been done on two different datasets. Most methods, including the proposed method, get good results on the ISOT dataset, which includes long text. However, other methods get poor results on FA-KES, while the proposed model has a good result. It can be concluded that the proposed method has the ability to work on different types of datasets. It also shows that the theory of the proposed model, which is simultaneously extracting spatial, temporal, and global features of text and using them for classification, works well.

The limitation of our work is that it is tested on English datasets only. Therefore, our future research will focus on adopting non-English datasets in order to build a more comprehensive system.

Our future work will also include considering alternative data sources, including social media and user-generated content in diverse languages. While these sources may present challenges such as interference and bias, they can provide valuable insights into the dissemination and consequences of fake news in virtual societies. Also, the study suggests exploring novel methodologies to further improve the precision of fake news detection systems. One potential avenue is the use of transfer learning techniques that leverage pre-existing language models, which have shown exceptional performance in various natural language processing tasks. Investigating the impact of different categories of fake news, such as propaganda, satire, and clickbait, on the efficacy of identification mechanisms could also be explored to enhance system efficiency, which is another future work.

Overall, the study highlights the potential of deep neural network architectures, along with feature extraction techniques and diverse datasets, in effectively detecting fake news.

REFERENCES

[1] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using text classification," *Secur. Privacy*, vol. 1, no. 1, p. 9, Jan. 2018.
[2] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018.
[3] V. L. Rubin, Y. Chen, and N. K. Conroy, "Deception detection for news: Three types of fakes," *Proc. Assoc. Inf. Sci. Technol.*, vol. 52, no. 1, pp. 1–4, Jan. 2015.
[4] G. Li, P. Niu, X. Duan, and X. Zhang, "Fast learning network: A novel artificial neural network with a fast learning speed," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1683–1695, Jun. 2014.
[5] W. Han and V. Mehta, "Fake news detection in social networks using machine learning and deep learning: Performance evaluation," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Nov. 2019, pp. 375–380.
[6] A. Jain, A. Shakya, H. Khatter, and A. K. Gupta, "A smart system for fake news detection using machine learning," in *Proc. Int. Conf. Issues Challenges Intell. Comput. Techn. (ICICT)*, vol. 1, Sep. 2019, pp. 1–4.
[7] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100007.
[8] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106983.
[9] E. Elsaeed, O. Ouda, M. M. Elmogy, A. Atwan, and E. El-Daydamony, "Detecting fake news in social media using voting classifier," *IEEE Access*, vol. 9, pp. 161909–161925, 2021.
[10] H. Saleh, A. Alharbi, and S. H. Alsamhi, "OPCNN-FAKE: Optimized convolutional neural network for fake news detection," *IEEE Access*, vol. 9, pp. 129471–129489, 2021.
[11] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," *Multimedia Tools Appl.*, vol. 80, no. 8, pp. 11765–11788, Mar. 2021.
[12] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti, and M. Akbar, "Fake news detection using deep learning models: A novel approach," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, p. e3767, Feb. 2020, doi: 10.1002/ett.3767.
[13] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "FNDNet—A deep convolutional neural network for fake news detection," *Cognit. Syst. Res.*, vol. 61, pp. 32–44, Jun. 2020.
[14] M. K. Elhadad, K. F. Li, and F. Gebali, "A novel approach for selecting hybrid features from online news textual metadata for fake news detection," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*. Cham, Switzerland: Springer, 2020, pp. 914–925.
[15] L. Borges, B. Martins, and P. Calado, "Combining similarity features and deep representation learning for stance detection in the context of checking fake news," *J. Data Inf. Qual.*, vol. 11, no. 3, pp. 1–26, Sep. 2019.

[16] B. Ghanem, P. Rosso, and F. Rangel, "Stance detection in fake news a combined feature representation," in *Proc. 1st Workshop Fact Extraction Verification*, Brussels, Belgium, 2018, pp. 66–71.

[17] A. Hanselowski, P. V. S. Avinesh, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance-detection task," in *Proc. 27th Int. Conf. Comput. Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 1859–1874.

[18] Q. Liao, H. Chai, H. Han, X. Zhang, X. Wang, W. Xia, and Y. Ding, "An integrated multi-task model for fake news detection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5154–5165, Nov. 2022.

[19] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Marquez, and A. Moschitti, "Automatic stance detection using end-to-end memory networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, New Orleans, LA, USA, 2018, pp. 767–776.

[20] C. Dulhanty, J. L. Deglint, I. B. Daya, and A. Wong, "Taking a stance on fake news: Towards automatic disinformation assessment via deep bidirectional transformer language models for stance detection," 2019, *arXiv:1911.11951*.

[21] M. Umer, Z. Imtiaz, S. Ullah, A. Mehmood, G. S. Choi, and B.-W. On, "Fake news stance detection using deep learning architecture (CNN-LSTM)," *IEEE Access*, vol. 8, pp. 156695–156706, 2020.

[22] T. Jiang, J. P. Li, A. U. Haq, A. Saboor, and A. Ali, "A novel stacking approach for accurate detection of fake news," *IEEE Access*, vol. 9, pp. 22626–22639, 2021.

[23] A. Choudhary and A. Arora, "Linguistic feature based learning model for fake news detection and classification," *Exp. Syst. Appl.*, vol. 169, May 2021, Art. no. 114171.

[24] M. Choudhary, S. S. Chouhan, E. S. Pilli, and S. K. Vipparthi, "BerConvoNet: A deep learning framework for fake news classification," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107614.

[25] A. J. Keya, M. A. H. Wadud, M. F. Mridha, M. Alatiyyah, and M. A. Hamid, "AugFake-BERT: Handling imbalance through augmentation of fake news using BERT to enhance the performance of fake news classification," *Appl. Sci.*, vol. 12, no. 17, p. 8398, Aug. 2022.

[26] M. I. Nadeem, S. A. H. Mohsan, K. Ahmed, D. Li, Z. Zheng, M. Shafiq, F. K. Karim, and S. M. Mostafa, "HyproBERT: A fake news detection model based on deep hypercontext," *Symmetry*, vol. 15, no. 2, p. 296, Jan. 2023, doi: 10.3390/sym15020296.

[27] F. W. R. Tokpa, B. H. Kamagaté, V. Monsan, and S. Oumtanaga, "Fake news detection in social media: Hybrid deep learning approaches," *J. Adv. Inf. Technol.*, vol. 14, no. 3, pp. 606–615, 2023.

[28] M. Asgari-Chenaghlu, M.-R. Feizi-Derakhshi, L. Farzinvash, M.-A. Balafar, and C. Motamed, "Topic detection and tracking techniques on Twitter: A systematic review," *Complexity*, vol. 2021, pp. 1–15, Jun. 2021.

[29] F. Ebrahimzadeh et al., "A hybrid recurrent neural network approach for detecting abnormal user behavior in social networks," Res. Square, Aug. 2023, doi: 10.21203/rs.3.rs-3242416/v1.

[30] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, 2016, pp. 207–212.

[31] N. Cecillon, V. Labatut, R. Dufour, and G. Linarès, "Abusive language detection in online conversations by combining content- and graph-based features," *Front. Big Data*, vol. 2, p. 8, Jun. 2019.

[32] S. Lu, M. Liu, L. Yin, Z. Yin, X. Liu, and W. Zheng, "The multi-modal fusion in visual question answering: A review of attention mechanisms," *PeerJ Comput. Sci.*, vol. 9, p. e1400, May 2023.

[33] S. Lu, Y. Ding, M. Liu, Z. Yin, L. Yin, and W. Zheng, "Multiscale feature extraction and fusion of image and text in VQA," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, p. 54, Apr. 2023.

[34] University of Victoria. (2021). *Fake News Detection Datasets*. [Online]. Available: https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/fake-news-detection-datasets

[35] F. K. A. Salem, R. A. Feel, S. Elbassuoni, M. Jaber, and M. Farah, "FA-KES: A fake news dataset around the Syrian war," in *Proc. Int. AAAI Conf. Web Social Media*, 2019, pp. 573–582. [Online]. Available: https://zenodo.org/record/2607278#ZBMr4RRBxPY

AHMED HASHIM JAWAD ALMARASHY received the B.S. degree in control and systems engineering from the University of Technology, Baghdad, Iraq, in 2003, and the M.Tech. degree in computer science engineering from Acharya Nagarjuna University, India, in 2016. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, University of Tabriz, Iran. His research interests include natural language processing, deep learning, image processing, and smart cities.

MOHAMMAD-REZA FEIZI-DERAKHSHI received the B.S. degree in software engineering from the University of Isfahan, Iran, and the M.Sc. and Ph.D. degrees in artificial intelligence from the Iran University of Science and Technology, Tehran, Iran. He is currently a Professor with the Faculty of Computer Engineering, University of Tabriz, Iran. His research interests include natural language processing, optimization algorithms, deep learning, social network analysis, and intelligent databases.

PEDRAM SALEHPOUR (Member, IEEE) received the B.S. and M.Sc. degrees in computer science and the Ph.D. degree in electrical engineering from the University of Tabriz, in 2007, 2009, and 2015, respectively. He is currently an Assistant Professor with the Faculty of Electrical and Computer Engineering, University of Tabriz. His research interests include distributed computing, machine learning, image processing, and deep learning.