IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 March 2024, accepted 20 March 2024, date of publication 25 March 2024, date of current version 29 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3381038

RESEARCH ARTICLE

Advancing Fake News Detection: Hybrid Deep Learning With FastText and Explainable AI

EHTESHAM HASHMI[1], SULE YILDIRIM YAYILGAN[1], MUHAMMAD MUDASSAR YAMIN[1], SUBHAN ALI[2], AND MOHAMED ABOMHARA[1]

[1]Department of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway
[2]Department of Computer Science (IDI), Norwegian University of Science and Technology (NTNU), 2815 Gjøvik, Norway

Corresponding author: Ehtesham Hashmi (hashmi.ehtesham@ntnu.no)

This work was supported by the Research Council of Norway through the SOCYTI Project (Technological Convergence Related to Enabling Technologies) under Grant 331736.

ABSTRACT The widespread propagation of misinformation on social media platforms poses a significant concern, prompting substantial endeavors within the research community to develop robust detection solutions. Individuals often place unwavering trust in social networks, often without discerning the origins and authenticity of the information disseminated through these platforms. Hence, the identification of media-rich fake news necessitates an approach that adeptly leverages multimedia elements and effectively enhances detection accuracy. The ever-changing nature of cyberspace highlights the need for measures that may effectively resist the spread of media-rich fake news while protecting the integrity of information systems. This study introduces a robust approach for fake news detection, utilizing three publicly available datasets: WELFake, FakeNewsNet, and FakeNewsPrediction. We integrated FastText word embeddings with various Machine Learning and Deep Learning methods, further refining these algorithms with regularization and hyperparameter optimization to mitigate overfitting and promote model generalization. Notably, a hybrid model combining Convolutional Neural Networks and Long Short-Term Memory, enriched with FastText embeddings, surpassed other techniques in classification performance across all datasets, registering accuracy and F1-scores of 0.99, 0.97, and 0.99, respectively. Additionally, we utilized state-of-the-art transformer-based models such as BERT, XLNet, and RoBERTa, enhancing them through hyperparameter adjustments. These transformer models, surpassing traditional RNN-based frameworks, excel in managing syntactic nuances, thus aiding in semantic interpretation. In the concluding phase, explainable AI modeling was employed using Local Interpretable Model-Agnostic Explanations, and Latent Dirichlet Allocation to gain deeper insights into the model's decision-making process.

INDEX TERMS Fake news, deep learning, interpretability modeling, machine learning, word embeddings, transformers.

I. INTRODUCTION

In the current era, digital platforms such as social media, online forums, and websites have overtaken traditional media as the foremost sources of information [1]. This paradigm shift highlights the transformation in our methods of accessing and interacting with information [2]. Social media's freedom of expression and instant information make it very popular, especially with the younger generation. People all over the world use these platforms to get news about everything from celebrities to politics, often without questioning if the news is real or not [3]. Fake news, which is intentionally created and verifiably false information, is seen as a threat to the stability of democratic systems, diminishing public trust in government institutions, and having a profound effect on critical societal aspects such as elections, economic conditions, and public opinions on matters like wars [4], [5]. The dissemination of fake news was markedly prominent in the key stages of the 2016 U.S. presidential election. This trend not only influenced public perception but also

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

raised concerns about the integrity of information consumed by voters during such significant democratic processes [6]. During that period, around 19 million bot accounts were established to disseminate false news regarding Trump and Clinton and this deliberate strategy rapidly increased the spread and influence of misinformation among the public [7], [8]. Additionally, reports indicate that fake news tends to receive more attention on social media compared to factual news, with examples of this trend visible on prominent social media platforms. The issue of fake news is considered to be more critical than other types of misinformation [9], [10]. As the widespread presence of fake news on social media continues to challenge the trustworthiness of online information, it becomes increasingly important to develop effective measures to address this problem. With the continuous increase in data volume, the need to rapidly and efficiently gather pertinent information becomes increasingly important. This underscores the importance of using computational linguistic methods. In this context, the application of Artificial Intelligence (AI) techniques becomes crucial, providing advanced tools to detect and address misinformation effectively.

The use of AI in fake news detection is critical because it can methodically analyze the minute details of language and context that might be missed by human moderators [11], [12]. Recent progress in AI and Natural Language Processing (NLP) has heightened the interest in fake news detection, resulting in the creation of many innovative approaches for research in this area [13], [14]. The extensive array of online content, encompassing a wide range of subjects, increases the complexity of the task. This has led researchers to focus on developing methods for automated detection of fake news. Consequently, this advancement in technology is crucial for maintaining the integrity of information on the internet [15]. Identifying fake news presents a significant technological challenge for several reasons. This complexity necessitates advanced solutions to ensure the reliability and accuracy of information disseminated online. This paper utilizes Machine Learning (ML) and Deep Learning (DL) based techniques, including state-of-the-art transformer-based models, to enhance fake news detection. By incorporating FastText word embeddings for effective text data processing and applying these methods to three publicly available datasets, we achieve a thorough and detailed analysis. This approach is crucial for accurately identifying misinformation in the world of online media. Additionally, our work integrates explainable AI methods, ensuring that our processes are not only effective but also transparent and understandable, aligning with the growing need for accountability in AI-driven solutions.

These advanced DL-based models are excellent when it comes to classification, but these models operate as black boxes [16]. To understand how the model works and which attributes contribute most to a prediction, Explainable AI (XAI) comes into play. In this work, we have utilized XAI algorithms to determine the words that contributed the most to the classification of a sentence as fake news. We have employed LIME with multiple deep learning models to interpret these black-box deep learning models.

Following, the contributions of this research work are summarized, followed by how the rest of the paper is organized.

A. WORK CONTRIBUTION

1) In this study, our focus is on advancing the detection of fake news by the refinement and application of established fake news detection methodologies through the use of regularization methods, optimization techniques, and hyperparameter tuning. Our methodology is carefully applied to a baseline dataset suited for binary classification, differentiating between factual and fabricated information. We carried out our work using three publicly available fake news datasets: WELFake, and two other news article datasets from Kaggle.
2) We stacked supervised and unsupervised FastText embeddings into ML-based models, including Support Vector Machine (SVM), Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), and bagging classifiers like Extreme Gradient Boosting (XGBoost), and Categorical Boosting (CATBoost). To ensure comprehensive coverage of text data, we also implemented a solution to handle out-of-vocabulary (OOV) words using FastText embeddings, allowing our models to effectively process previously unseen terms. In addition, we pursued rigorous optimization, fine-tuning regularization techniques and hyperparameters across our ML models. This meticulous approach aimed to optimize model performance, prevent overfitting, and ultimately produce robust, generalizable results.
3) Additionally, to effectively capture complex contextual information and sequential dependencies within the text data, we applied FastText embeddings in DL-based models such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN). Furthermore, this study implemented state-of-the-art text classification transformer-based models, including Bidirectional Encoder Representation from Transformers (BERT), Robustly Optimized BERT (RoBERTa), and the auto-regressive transformer XLNET with hyperparameter tuning. We leveraged these transformers for their proven ability to capture intricate contextual information and long-range dependencies in text data, making them well-suited for the complex task of fake news detection.
4) To enhance the interpretability of our results, particularly after observing the best performance of the CNN-LSTM model, we implemented Explainable AI (XAI) techniques. These included Local Interpretable Model-Agnostic Explanations (LIME) and coupled

with topic modeling using Latent Dirichlet Allocation (LDA), all applied to the WELFake dataset.

B. STRUCTURE OF THE PAPER

The structure of the remainder of this paper is organized as follows: Section (II) reviews the existing research on fake news detection. Section (III) details the methodology of the proposed work. Section (IV) is dedicated to presenting the results and discussions. Section (V) compares these results with baseline methods. Section (VI) delves into the interpretability modeling using LIME, and LDA. In the concluding phase, Section (VII) concludes the paper and outlines future work.

II. RELATED WORK

In this section, we discuss the existing research in the field of fake news detection, where extensive studies have explored various methodologies ranging from traditional ML and DL to transformer-based methods.

A. ML BASED APPROACHES

Choudhury and Acharjee [17] proposed an ML-based approach for fake news detection using three different datasets: Liar [18], Fake Job Posting [19], and Fake News. After data pre-processing, the cleaned text was then converted into numerical features using Term Frequency Inverse Document Frequency (TF-IDF) to select the categorical features, and these features were then fed to various ML-based algorithms, including Naive Bayes (NB) [20], SVM [21], LR [22], and RF [23]. The SVM classifier achieved the highest accuracy with 61%, 97%, and 96% in these datasets, respectively. Altheneyan and Alhadlaq [14] introduced a distributed ML-based approach for fake news detection using the Spark framework [24]. Their study utilized the False News Challenge (FNC-1) dataset, categorizing fake news into four distinct categories. Leveraging big data technology with Spark, they assessed and compared their method with other state-of-the-art approaches. Their approach involved creating a stacked ensemble model and experimenting on a distributed Spark cluster. To enhance performance, they explored three distinct word embedding techniques: N-grams [25], Hashing TF-IDF, and Count Vectorizer (CV) [26]. Akhtar et al. [27] introduced a query expansion technique for detecting fake news and disinformation with the integration of AI and ML, aiming to mitigate Supply Chain Disruptions (SCD). They focused on four prominent Pakistani online news sources: 'Geo News,' 'The Dawn,' 'Express Tribune,' and 'The News.' Their study involved analyzing approximately 500 pages from each source to extract relevant events and topics spanning from January to April 2021. The SCD data were categorized into various types, including natural, human-caused, maritime, and mass disruptions, all associated with fake news and disinformation.

Shalini et al. [28] proposed ML-based techniques to distinguish between bot-generated and human-generated information on social media. The process entails extracting characteristics from a dataset based on phrase frequency and then applying classification algorithms. The method is particularly effective at detecting rogue accounts within biassed datasets, which are typical in social media platforms. The technology distinguishes between legitimate and fake identities with high accuracy. The system achieves improved accuracy by utilizing Recurrent Neural Networks (RNNs) with multiple activation functions. Furthermore, as the number of folds in cross-validation increases, the classification precision improves. The experimental analysis includes tests on both synthetic and real-time social media datasets, with real-time Twitter data obtaining roughly 96% accuracy and synthetic datasets achieving 98% accuracy.

B. DL AND TRANSFORMER BASED APPROACHES

Verma et al. [29] presented Word Embedding Over Linguistic Features for Fake News Detection (WELFake) a novel two-phase benchmark model to authenticate news content by leveraging machine learning classification with word embedding over linguistic features. This comprehensive approach demonstrates a remarkable improvement in fake news detection, with the WELFake model achieving a peak accuracy of 96.73%. This performance surpasses traditional methods, including BERT and CNN models, by up to 4.25%, highlighting the efficacy of combining linguistic features with advanced embedding techniques. The study further contributes a novel dataset comprising approximately 72,000 articles, enhancing the model's reliability and generalizability across diverse datasets. Shu et al. [30] introduced FakeNewsNet, a repository designed to support research on fake news detection on social media. This repository comprises two detailed datasets, rich in news content, social context, and spatiotemporal information, to overcome the limitations of existing datasets. The comprehensive analysis of FakeNewsNet sheds light on its potential applications in detecting fake news, aiming to address the challenges posed by the scarcity of multifaceted fake news datasets. This initiative marks a significant step towards enhancing the accuracy and effectiveness of fake news detection mechanisms.

C. Truică and Apostol [31] introduced an innovative approach that employs document embeddings to construct multiple models capable of accurately classifying news articles as either reliable or fake. Their evaluation encompassed various machine learning (ML) models, including NB, Gradient Boosting, DL-based models like LSTM and GRU, as well as three transformer-based models: pre-trained BERT [32], (Bidirectional and Auto-Regressive Transformers) BART [33], and RoBERTa [34] methods. These evaluations were conducted using five distinct datasets containing fake news articles, employing various word embeddings, including TF-IDF, WORD2VEC [35], and FastText [36]. In their study, Nanade and Kumar [37] proposed a transformer-based method for Twitter fake news detection using the BERT base model, which provided them with an accuracy score of 77.29%. Verma et al. [38] introduced

a binary classification framework for fake news detection that combines Bidirectional Encoder Representations from Transformers (BERT) to capture global text semantics through the relationships between words in sentences, and CNN to leverage N-gram features for local text semantics. They conducted their experiments on four publicly available datasets. A similar approach was proposed by Guo et al. [39] using DL-based models and a pre-trained transformer-based BERT model for the same purpose. The results of both studies provide valuable insights into the effectiveness of these methods in the domain of fake news detection.

Praseed et al. [40] presented an approach for detecting fake news in Hindi using an ensemble of pre-trained transformer models XLM-RoBERTa [41], mBERT, and ELECTRA [42] which are separately fine-tuned for the task of Hindi fake news detection. After undergoing appropriate fine-tuning, pre-trained transformer models have demonstrated their capability to identify fake news across various languages. In their research study, they utilized the CONSTRAINT2021 dataset [43], which comprises a total of 8192 online posts. Among these posts, 4358 are categorized as non-hostile, whereas the remaining 3834 posts exhibit some form of hostility. In their research study, Biradar et al. [44] introduced an early fusion-based approach that combined essential features extracted from context-based embeddings like BERT, XLNet, and ELMo [45]. This fusion method aimed to improve the collection of context and semantic information from social media posts, leading to increased accuracy in detecting false news. Alongside this approach, they implemented both ML and DL-based techniques. Their experiments were conducted using the ''CONSTRAINT shared task 2021'' dataset. Moreover, when considering the various embeddings discussed, BERT embeddings exhibited significantly superior performance compared to XLNet and ELMo, particularly when applied to the limited short text data extracted from Twitter. Additionally, combining features derived from different embeddings into a unified vector for classification resulted in a slight performance improvement.

Wu et al. [46] introduce Graph-based Semantic Structure Mining with Contrastive Learning (GETRAL), a revolutionary graph-based semantic structure mining framework with contrastive learning, to improve evidence-based fake news identification that significantly surpasses existing models on the Snopes [47] and PolitiFact [48] datasets. This methodology overcomes the constraints of earlier methods by representing claims and evidence as graph-structured data, allowing for the capture of long-distance semantic relationships. GETRAL lowers information redundancy through graph structure learning and enhances representation learning through supervised contrastive learning with adversarial augmented examples. On Snopes, GETRAL achieves an F1-Macro score of 80.61% and an F1-Micro score of 85.12%. On the PolitiFact dataset, GETRAL records an F1-Macro of 69.53% and an F1-Micro of 69.81%, demonstrating its superior performance in addressing the challenges of fake news detection by integrating advanced techniques for a more accurate and interpretable analysis. Soga et al. [49] focuses on the detection of fake news on social media by analyzing stance similarity and employing Graph Neural Networks (GNNs). Their research work proposes a method that accounts for the opinion similarity between users by examining their stances towards news articles and user post interactions. This method uses Graph Transformer Networks (GNNs) to extract both global structural information and interactions of similar stances effectively. The technique addresses stance analysis challenges in microblogs and minimizes the impact of poorly represented stance features. The approach was evaluated using custom crawled Twitter data and the benchmark FibVID[1] dataset, demonstrating significant improvements in detection performance compared to conventional methods, including state-of-the-art approaches. This advancement suggests that incorporating stance similarity in news-sharing interactions, alongside the extraction of propagation patterns characteristic of fake news, enhances the detection accuracy, making it a promising direction for future fake news detection studies. Pilkevych et al. [50] explored fake news detection by using GNNs, they did a detailed analysis aimed at mitigating the impacts of disinformation, particularly in the context of Russia's aggression against Ukraine. They advocate for GNNs as a potent tool for the automated identification of harmful content, emphasizing their application in monitoring online media to promptly detect and assess fake news. Their approach leverages knowledge graphs (KG) for entity recognition and relationship mapping in textual content, with an emphasis on detecting signs of negative psychological influence. Among the models evaluated, GraphSAGE stands out for its performance, achieving notable accuracy scores of 89.78% on the Politifact dataset and 98.01% on the Gossipcop dataset, when trained on data embodying signs of negative psychological influence. This research underscores the critical role of sophisticated machine learning techniques in addressing the challenge of disinformation, highlighting the effectiveness of GNNs in enhancing the accuracy and efficiency of fake news detection systems.

C. MAJOR CHALLENGES

After performing comprehensive analysis or related work following are the current challenges in fake news detection,

1) **Variability and Sophistication:** Fake news often mimics genuine news in style and presentation, making it difficult to distinguish based on surface features alone. The sophistication of misinformation tactics evolves continuously, necessitating advanced detection techniques that can adapt to changing patterns [51].
2) **Linguistic Nuances and Contextual Understanding:** The effective detection of fake news requires a deep understanding of linguistic subtleties and the ability to interpret context. This is challenging due to the vast

[1] https://github.com/merry555/FibVID

TABLE 1. Comparative analysis of selected studies.

Ref	Dataset	Feature Set	Method	Results
[14]	FNC[2]	N-grams, CV, TF-IDF, GloVe	DT, SVM, LR, RF, Ensembling	F1-score: 0.92
[17]	Liar [18], Fake Job Posting [19]	TF-IDF	NB, SVM, LR, RF	Accuracy: 0.96
[27]	Online News Channels	Query Expansion	SVM	Mean Reciprocal Rank (MRR): 0.65
[28]	Online Tweets	Lemma-Based, Rule-based, Hybrid	SVM, RF, NB, RNN	Accuracy: 0.96
[31]	Liar [18], Buzz Feed, Kaggle	TF-IDF, WORD2VEC, FastText	NB, Gradient Boosting, LSTM, GRU, BERT, BART, RoBERTa	Accuracy: 0.99
[37]	Online Tweets	Contextual Embeddings	BERT Base	Accuracy: 0.77
[38]	WELFake, FakeNews, Kaggle	N-grams, Contextual Embeddings	CNN, LR, NB, DT, RF, XGBoost, BERT	Accuracy: 0.99
[39]	Weibo[3]	One Hot Encoding	CNN, LSTM, BERT	F1-score: 0.89
[43]	CONSTRAINT Shared Task-2021	Contextual Embeddings	mBERT, XLM-RoBERTa, ELECTRA	MCC: 0.688
[44]	CONSTRAINT Shared Task-2021	Contextual Embeddings	BERT, XLNet, and ELMo	Accuracy: 0.97
[46]	Snopes [47], PolitiFact [48]	Contextual Embeddings	KG with GETRAL	F1-score: 0.81
[50]	Politifact, GossipCop	Contextual Embeddings	GCN, GAT, GraphSAGE	Accuracy: 0.98

[2]https://www.kaggle.com/datasets/abhinavkrjha/fake-news-challenge?select=train_stances.csv
[3]https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DULFFJ

diversity of languages and the specific cultural contexts within which news is disseminated [52].

3) **Bias and Subjectivity:** Identifying biases and subjective assertions within news content without suppressing freedom of expression or introducing detection biases presents a significant challenge.
4) **Scalability and Generalizability:** The ability to scale detection mechanisms to process vast quantities of data across different platforms, and ensuring these mechanisms are generalizable across various domains and languages, is a complex endeavor.

From the existing literature, it is evident that numerous studies have tackled the problem of fake news detection utilizing both traditional ML and DL-based approaches and highlight the current challenges in the domain of fake news detection, such as the sophisticated techniques used to generate and disseminate fake news, the rapid evolution of misinformation, and the difficulty of achieving high accuracy in detection while maintaining interpretability and generalizability. In this study, we aim to contribute to this analysis by employing a comprehensive range of techniques, including ML, DL, and transformer-based models. To enhance the accuracy and generalizability of fake news detection, we leverage supervised and unsupervised FastText word embeddings using three benchmark datasets, complemented by extensive regularization techniques and hyperparameter tuning methods. A noteworthy aspect of our contribution to this paper will be our focus on addressing the limited body of work concerning XAI within fake news. By incorporating explainable AI and topic modeling techniques into our research methodology, we intend to shed light on the interpretability and transparency of our models, ultimately enhancing the comprehensibility of fake news understanding. Table 1 represents the comparative analysis of the current state-of-the-art methods.

III. WORK METHODOLOGY

The proposed research methodology of this study involves a systematic approach to achieving promising results, as shown in Figure 1. Each of the steps from our research methodology is further elaborated in detail below:

A. DATASET

In our study, we addressed the binary classification problem, where 0 represents fake news, and 1 represents real news. We employed three publicly available datasets: WELFake [29], FakeNewsNet [30], and FakeNewsPrediction.[4]WELFake consists of 72,134 news articles, with 35,028 categorized as fake news and 37,106 classified as real news. To prevent classifier overfitting and enhance machine learning training, the authors combined data from four prominent news datasets, including those from Kaggle, McIntire, Reuters, and BuzzFeed Political, thereby enriching the dataset with a more extensive and varied collection of text data. FakeNewsNet comprises two extensive datasets that encompass a wide range of characteristics related to

[4]https://www.kaggle.com/datasets/rajatkumar30/fake-news

TABLE 2. Count of instances in datasets.

Dataset	Count of Instances
WELFake	72,134
FakeNewsNet	23,196
FakeNewsPrediction	6,335

news content, social context, and spatiotemporal information. The third dataset, FakeNewsPrediction, comprises 3,171 instances of real news and 3,164 instances of fake news. Table 2 represents the count of instances in three datasets used in this paper.

B. DATA PREPROCESSING

Effective data preprocessing plays a pivotal role in enhancing the performance of various ML and DL-based models, as it involves eliminating irrelevant text from the dataset and ensuring that the data is presented in a concise and suitable format. In our study, we placed particular emphasis on two primary columns: ''text,'' which contained all the news comments, and ''label,'' representing the true or fake label. The rationale behind text preprocessing lies in its ability to significantly impact the performance of learning algorithms. By preparing the data appropriately, we can improve the quality and relevance of information used for training and analysis. To preprocess the ''text'' column, we implemented a series of essential steps. Initially, we converted all uppercase letters to lowercase and removed non-essential characters, such as ASCII symbols. Subsequently, we conducted tokenization of both words and sentences while eliminating stop words to further refine the data. Moreover, we employed Python's RegEx library to filter and process elements such as numbers, punctuation, and specific patterns, including email addresses, URLs, and phone numbers. Additionally, we addressed the removal of duplicate examples within the dataset, ensuring data quality and diversity for model training. Data preprocessing ensures that the dataset is cleansed of extraneous information that might otherwise hinder the learning process. In addition to these steps, we applied lemmatization to our text data. Lemmatization is employed to reduce words to their base or root form, promoting consistency in word usage and improving the model's ability to recognize similarities between different inflections of the same word. Overall, our text preprocessing pipeline was designed to optimize the quality and relevance of the data fed into our learning algorithms, thereby enhancing the accuracy of fake news detection.

For our transformer-based models, we have streamlined our preprocessing to include word and sentence tokenization, converting uppercase characters to lowercase, and removing extraneous symbols. This focused approach is instrumental in addressing the issue of syntactic ambiguity, as highlighted in prior research [53]. Syntactic ambiguity presents a substantial challenge encountered in previous ML and DL-based algorithms, where words within a sentence can have multiple meanings depending on the context, making interpretation a complex endeavor. Table 3 highlights some preprocessed text data examples from the WELFake dataset.

C. WORD EMBEDDING

Word embeddings provide numerical representations for textual inputs, allowing machines to process and understand textual data more effectively. These embeddings capture semantic relationships and contextual information, facilitating tasks such as sentiment analysis, text classification, and language modeling. By transforming words into vectors in a continuous vector space, word embeddings enable machines to recognize similarities between words, capture word meanings, and generalize from the training data, ultimately enhancing the performance of various natural language tasks. In this paper, we have utilized FastText embeddings due to their effectiveness in capturing semantic information and contextual nuances within text data. FastText embeddings offer distinct advantages over traditional word embeddings, as they can represent subword information and handle out-of-vocabulary words more gracefully. These qualities make FastText embeddings a superior choice, particularly when dealing with languages with rich morphological structures and variations. Conventional word vectors disregard the internal structure of words, which holds valuable information. This information could prove beneficial when generating representations for infrequent or incorrectly spelled words. The equation 1 denotes the mathematical formula to compute FastText word embeddings [54].

$$u_w + \frac{1}{|N|}\sum_{n\in N} x_n \tag{1}$$

where:

u_w: represents the vector for a word w in the embedding space.

$\frac{1}{|N|}$: is the fraction representing the average.

$\sum$: is the sum symbol, used to sum over a set of vectors.

$n \in N$: specifies that we are summing over the set N.

x_n: represents the vector for the context words in the set.

FastText, a word representation tool developed by Facebook's research division, provides both unsupervised and supervised modes, featuring an extensive lexicon of 2 million words sourced from Common Crawl. Each word is represented in a 300-dimensional vector space, resulting in a vast library comprising a staggering 600 billion word vectors. What sets this word embedding method apart is its unique approach, incorporating manually crafted n-grams as features in addition to individual words [55]. FastText offers two primary modes of usage: unsupervised and supervised. In our research, we have employed both of these modes, conducting a comprehensive analysis of their respective applications.

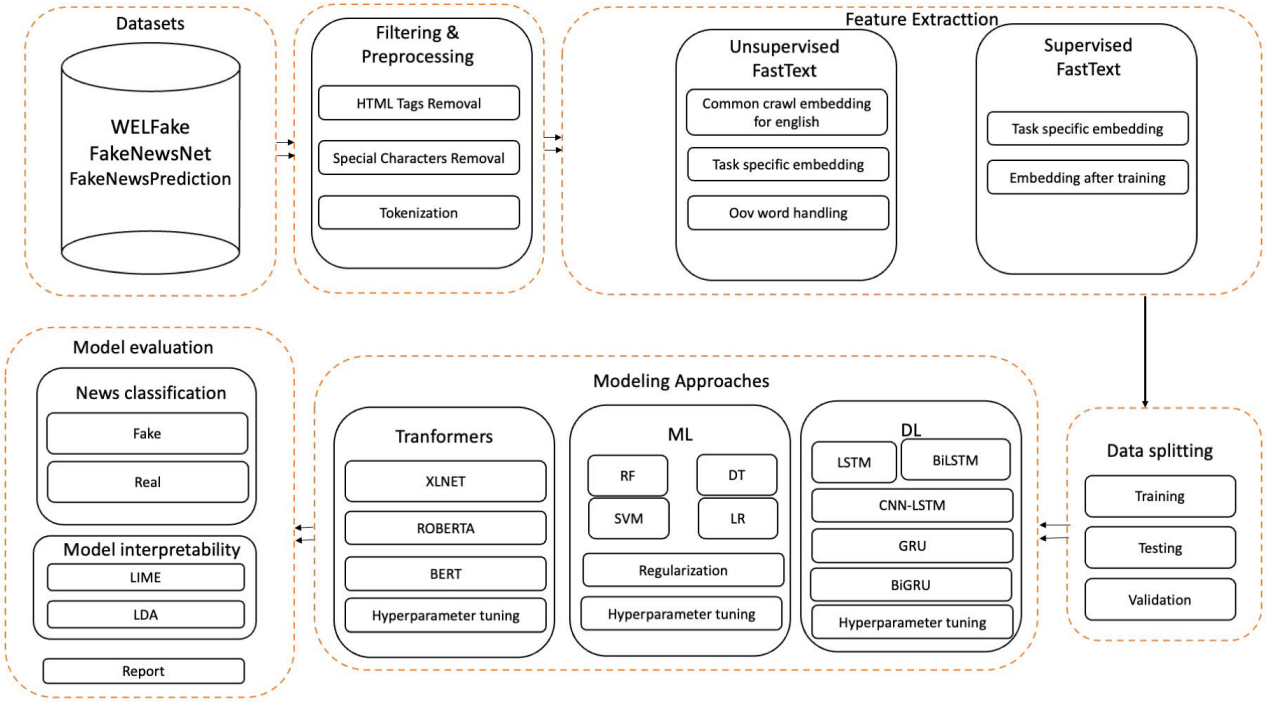

FIGURE 1. Methodology diagram.

TABLE 3. Examples of preprocessed data on WELFake dataset.

Class Label	Original Text	Preprocessed Text
1	DR. BEN CARSON TELLS THE STORY OF WHAT HAPPENED WHEN HE SPOKE OUT AGAINST OBAMA:	ben carson tells the story of what happened when he spoke out against obama
0	Killing Obama administration rules, dismantling Obamacare and pushing through tax reform are on the early to-do list.	killing obama administration rule dismantling Obamacare pushing tax reform early list
1	source Add To The Conversation Using Facebook Comments	source add to the conversation using facebook Comments
0	Notable names include Ray Washburne (Commerce), a Dallas-based investor, is reported to be under consideration to lead the department.	notable name include ray washburne commerce dallas based investor investor reported consideration lead department

1) UNSUPERVISED FASTTEXT

In unsupervised learning, FastText generates word vectors, extending the Word2Vec model to include subword information by breaking words into a bag of character n-grams. For example, with the word ''Obama'', FastText would consider not just ''Obama'' but also n-grams like ''Oba'', ''bam'', ''ama'', depending on the specified n-gram range. Similarly, for ''Trump'', it would analyze fragments like ''Tru'', ''rum'', ''ump''. This approach is valuable for understanding suffixes and prefixes, helping the model recognize that words with similar subparts might be semantically related. In FastText's approach to unsupervised learning, when breaking down words into a bag of character n-grams, the typical range for these n-grams is between 3 to 6 characters. FastText's unsupervised learning method uses vast amounts of unlabeled text to build word representations. These word vectors can be utilized in various tasks such as word similarity, word analogy, or as features in downstream NLP applications.

In this study, we used FastText's unsupervised word vectors, specifically the pre-trained model cc.en.300.bin.[5] This model was developed on Common Crawl and Wikipedia using FastText's unsupervised learning technique, which integrates subword information into the training process. By doing so, the model captures the morphological intricacies of words and represents them as vectors in a 300-dimensional space. Each word vector is enhanced with information collected from character n-grams, helping the model to better understand word morphology and handle out-of-vocabulary terms. The introduction of subword information allows our study to include not just the semantic representation

[5]https://fasttext.cc/docs/en/english-vectors.html

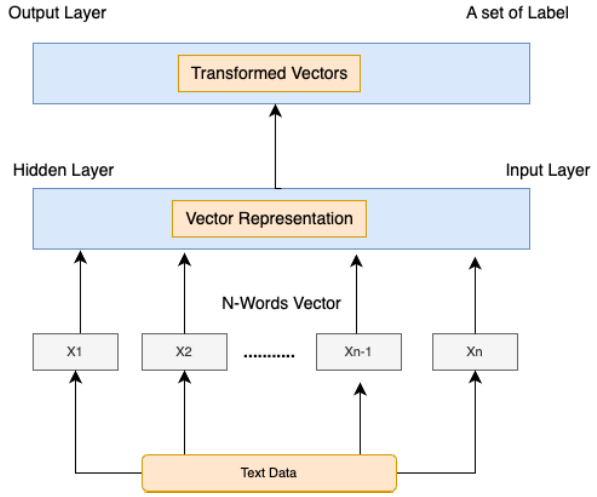

FIGURE 2. FastText word embedding architecture.

of complete words but also the semantic implications of their constituent pieces, providing a more sophisticated perspective of language semantics.

In our implementation, we use FastText's unsupervised model to create word embeddings for a Fake News dataset. The function *text_to_fasttext_embeddings* processes each text, generating embeddings for each word using get_word_vector. This method effectively handles OOV words by leveraging subword information. It computes the average of these embeddings to represent the entire text. If a text has no known words or is entirely OOV, a zero vector is returned. Applied to our dataset df, this approach results in a feature matrix *X_fasttext*, suitable for various analytical tasks in our Fake News study. Algorithm 1 explains the unsupervised FastText with *OOV* used in our paper.

2) SUPERVISED FASTTEXT

In supervised learning, FastText is used for text classification. It applies the same principle of using subword information but is trained on a labeled dataset where each text snippet has an associated label or category. FastText uses a hierarchical softmax function based on the Huffman coding tree which speeds up training and prediction time, making it feasible to train on millions of documents. For text classification, the model averages the word vectors in a text to form the text representation, which is then used to predict a label. FastText's supervised mode is particularly powerful because it can handle large datasets and large numbers of classes efficiently.

In this study, we did a thorough exploration by deploying both supervised and unsupervised FastText models. While both approaches produced encouraging results, it was clear that supervised FastText consistently beat and outperformed its unsupervised counterpart. This analysis emphasizes the importance of using labeled training data in text classification problems, where supervised learning can exploit explicit category information to obtain greater accuracy. The effectiveness of the supervised FastText model confirms its applicability for classification-oriented tasks and highlights its potential as a significant tool for improving the accuracy of our research. In our experimentation, we trained the FastText model over 50 epochs, employing learning rates of 0.01, 0.1, and 0.01 respectively for three different datasets. This training strategy allowed us to harness the power of FastText embeddings to enhance our classification performance effectively.

Algorithm 1 Create Unsupervised FastText Embeddings for Text Data

Require: FastText model file 'cc.lang.300.bin', text data from *df*

Ensure: Matrix *X_fasttext* of FastText embeddings

```
Load the FastText model: ft_model ← fasttext.load_model('cc.lang.300.bin');
function text_to_fasttext_embeddings(text, ft_model)
    words ← split the text;
    embeddings ← initialize an empty list;
    for each word in words do
        vector ← ft_model.get_word_vector(word);
        if vector is valid then
            Append vector to embeddings;
        end if
    end for
    if embeddings is not empty then return mean of embeddings across axis 0;
    else
        return zero vector of length ft_model.get_dimension();
    end if
end function
X_fasttext ← stack vertically the result of text_to_fasttext_embeddings for each text in df;
```

TABLE 4. Hyperparameters details for supervised FastText.

Dataset	Epoch	Learning Rate	test_size
WELFake	50	0.01	0.2
FakeNewsNet	50	0.1	0.2
FakeNewsPrediction	50	0.01	0.2

D. MODELING APPROACHES

This section will detail the ML, DL, and transformer-based models utilized in this paper. It will provide an in-depth examination of each model's architecture and its application within our research framework.

1) ML BASED MODELS

FastText embeddings were used as input for the subsequent supervised ML-based models, including DT, SVM, LR, and RF. In addition, boosting methods such as XGBoost and CatBoost, along with feature engineering techniques, were applied. In the implementation of ML-based models, several

hyperparameters and regularization techniques have been employed to optimize performance.

TABLE 5. Configuration details for ML models.

model	regularization	hyperparameter
DT	Split_{min}: [2, 5, 10]	GridSearchCV
RF	N-Estimators: [50, 100, 200]	GridSearchCV
SVM_{linear}	C: [0.1, 1, 10]	GridSearchCV
LR	C: [1, 10, 100]	GridSearchCV

In table 5, for DT, the **Split_min** values of 2, 5, and 10 dictate the minimum number of samples required for a node split, influencing the tree's complexity and potential overfitting. In RF, **the N-Estimators** parameter, with values 50, 100, and 200, determines the number of trees in the forest, balancing between computational efficiency and model accuracy. The SVM with a linear kernel and LR classifiers both utilize the regularization parameter **C**, tested at values 0.1, 1, and 10 for SVM, and 1, 10, and 100 for LR. The **C** parameter plays a crucial role in controlling the strength of regularization, which helps to prevent overfitting by penalizing the magnitude of the coefficients. Lower values of **C** imply more regularization, constraining the model to simpler decision boundaries.

All these parameters across different models were meticulously optimized using GridSearchCV, an exhaustive search over specified parameter values. GridSearchCV systematically evaluates combinations of parameters, selecting the ones that yield the best performance metrics, thereby ensuring that each model is finely tuned for optimal accuracy and generalization. The equation 2 represents the GridSearchCV algorithm in ML. In this formulation, *optimize* reflects the goal of **GridSearchCV** to find the best model parameters. The hyperparameters $h_1 \in H_1$, $h_2 \in H_2$, $\ldots$, $h_n \in H_n$ are exhaustively searched to maximize the *score* function within their ranges. The *argmax* operator identifies the specific set of hyperparameters that yield the highest score, typically a measure of model accuracy or performance.

$$optimize\left(\underset{h_1 \in H_1,\, h_2,\, \ldots,\, h_n \in H_n}{argmax} score\,(model(h_1, h_2, \ldots, h_n))\right) \tag{2}$$

The following table 6 represents the hyperparameters and regularization details for boosting methods in the proposed approach,

TABLE 6. Configuration details for boosting algorithms.

model	rounds	lr	depth	sample	function	Objective
XGBoost	100	0.1	3	0.8	logloss	logistic
CatBoost	100	0.1	6	0.7	logloss	binary
AdaBoost	100	1.0	None	None	None	binary

For the gradient boosting models, XGBoost and CatBoost, the configuration includes 100 boosting rounds, a learning rate of 0.1, and logloss as the loss function. XGBoost uses a maximum depth of 3 and a subsample ratio of 0.8, while CatBoost uses a maximum depth of 6 and a subsample ratio of 0.7. These parameters are critical in managing the models' complexity and preventing overfitting while ensuring efficient learning.

2) DL BASED MODELS

In our study, we implemented LSTM, its variant BiLSTM, GRU, and the hybrid CNN-LSTM model. These RNN-based models excel in processing sequential data, with LSTM units adept at capturing long-term dependencies. The BiLSTM variant further enhances this by processing data in both forward and backward directions, thus gaining a more comprehensive understanding of context, which is especially beneficial in complex sequential tasks. GRU, while similar to LSTM in managing sequence dependencies, offers a more streamlined architectural design. Additionally, the CNN-LSTM model combines CNN with LSTM, leveraging CNNs' ability to extract spatial features and LSTMs' strength in interpreting these features temporally. This hybrid model is particularly effective in tasks that require an understanding of both spatial and temporal patterns, such as video classification and time-series forecasting.

1) **Regularization Techniques:** Regularization techniques serve as a method in classifier training to avoid overfitting, a condition where a model predicts training data accurately but fails to generalize well to new, unseen data. The performance enhancement of the CNN-LSTM model is significantly attributed to the use of kernel L2 regularization, with a lambda setting of 0.01 for both LSTM and CNN layers. The importance of L2 regularization lies in its ability to minimize weight magnitudes, thereby encouraging the model to adopt smaller values for weights. This approach accomplishes two key goals: it minimizes the likelihood of overfitting and preserves the model's ability to generalize across different datasets effectively. The preference for L2 regularization over L1 was a calculated choice. L1 regularization, while capable of inducing sparsity by turning some weights to zero, could lead to underfitting, an issue that emerged in the initial testing phases. The formulas for $L1$ and $L2$ regularization are detailed in equations (3) and (4), respectively.

$$L1(\mathbf{w}) = \lambda \sum_{i=1}^{n} |w_i| \tag{3}$$

where:

$\mathbf{w}$: is the weight vector of the model

λ: is the regularization coefficient

n: is the number of weights in the vector

w_i: is the ith weight in the weight vector

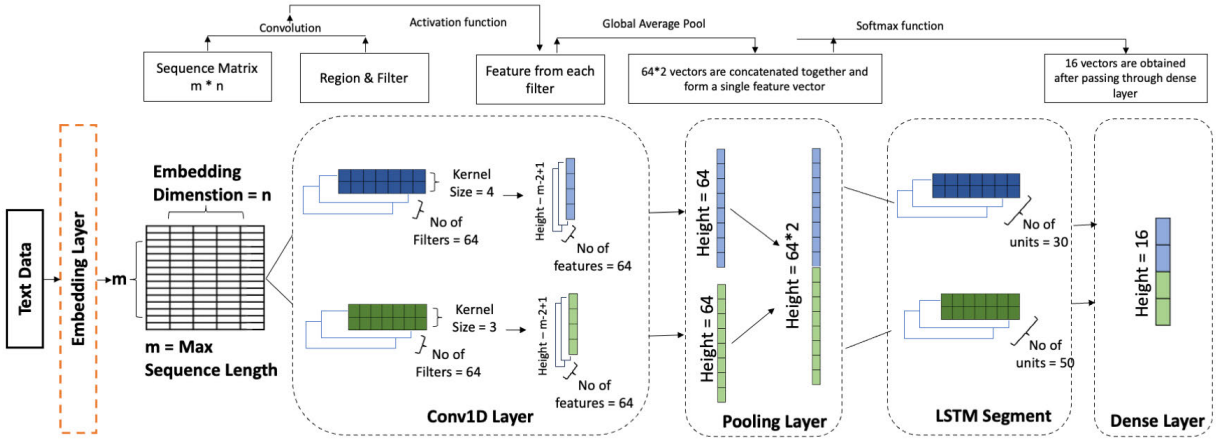

FIGURE 3. CNN-LSTM proposed model architecture diagram.

$L1$ regularization incorporates the absolute magnitude of coefficients as a penalty to the loss function. This addition of absolute values introduces a non-linear penalty based on the weights, making $L1$ regularization conducive to sparse outcomes where numerous coefficients become precisely zero.

$$L2(\mathbf{w}) = \lambda \sum_{i=1}^{n} w_i^2 \tag{4}$$

$L2$ regularization introduces the squared magnitude of coefficients as a penalty to the loss function. This squaring process results in a smoother, differentiable penalty, even at $w_i = 0$. Contrary to $L1$ regularization, $L2$ does not lead to sparse models because it generally does not push coefficients to become exactly zero, although it may reduce them to small values.

2) **Hyperparameter Tuning for DL-Based Models:** In the hyperparameter optimization process for DL-based models, we methodically adjusted the model's learning process through targeted experimentation. The training period was set to 10 epochs, a duration chosen to balance effective learning against the risk of overfitting, and ended when the model's loss decreased.

In figure 3, CNN-LSTM model combines two convolutional layers and LSTM layers for advanced data processing. The convolutional layers, each with 64 filters, use kernel sizes of 4 and 3 respectively, with 'relu' activation, effectively extracting spatial features. A MaxPooling layer follows, reducing data dimensionality and enhancing efficiency. The LSTM segment, with two layers of 50 and 30 units, captures temporal dynamics, crucial for sequential data analysis. The model concludes with a 'softmax'-activated dense layer, making it suitable for classification tasks. This architecture excels in tasks requiring both spatial feature extraction and temporal sequence understanding. Table 7 illustrates the hyperparameters and configuration details of each DL-based model. Notably, the count of each layer has been mentioned as well.

3) TRANSFORMER BASED MODELS

The Transformer, an innovative system in Natural Language Processing (NLP), is structured to handle sequence-to-sequence tasks, utilizing a self-attention mechanism that efficiently manages long-range dependencies comprising two main components encoder and decoder. BERT, RoBERTa, and XLNet are all encoder-only models. This architecture makes them highly effective for text classification tasks, where understanding and processing input data to generate contextual representations is crucial. Transformers were first introduced in 2017 by Vaswani et al. [56], the Transformer's self-attention mechanism is characterized by its ability to focus on different parts of the input sequence, which can be represented through a specific mathematical formulation.

$$Attention(Q, K, V) = \text{softmax}(\frac{QK_i^\top}{\sqrt{d_k}})V_i \tag{5}$$

where:

Q: is the loss to minimize
K: is the key matrix
V: is the value matrix
d_k: is the dimension of the key vectors
N: is the length of the input sequence
i: is the index of the query vector

This study concentrates on the use of transformers, with a particular emphasis on the optimization of their hyperparameters. Transformers represent a notable progression from earlier language models like RNNs, which were limited by their computational intensity and memory demands, especially in generative tasks. In our research, we leveraged extensive text datasets and utilized text classification transformers, including BERT, XLNet, and RoBERTa. BERT

TABLE 7. Configuration details for DL models.

model	model layer	dense layer	dropout layer	pooling layer	flatten layer	epochs	function	loss	optimizer
CNN-LSTM	4	2	2	2	1	10	softmax	categorical entropy	adam
LSTM	2	1	1	-	-	10	softmax	categorical entropy	adam
BiLSTM	2	1	1	-	-	10	softmax	categorical entropy	adam
GRU	2	1	1	-	-	10	softmax	categorical entropy	adam
BiGRU	2	1	1	-	-	10	softmax	categorical entropy	adam
BiLSTM-GRU	3	1	2	-	-	10	relu	categorical entropy	adam

TABLE 8. Configuration details for transformer based models.

model	tokenizer	batches	lr	epoch
BERT	BertTokenizer	32	2e-5	5
XLNet	AutoTokenizer	32	2e-3	5
RoBERTa	AutoTokenizer	32	2e-5	5

excels in understanding the context of a word in a sentence by looking at the words that come before and after it. XLNet, an extension of the Transformer model, outperforms BERT in certain scenarios by using a permutation-based training approach. RoBERTa modifies key hyperparameters in BERT, including removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates, leading to improved performance on several benchmarks. The table 8 represents the hyperparameters and configuration details for transformer-based methods in the proposed approach,

IV. RESULTS AND DISCUSSION

In our assessment, we utilized standard metrics to evaluate the model's performance. These metrics include accuracy, precision, recall, and F1-score, all of which offer quantitative measures of the model's effectiveness.

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{6}$$

$$\text{Precision} = \frac{T_P}{T_P + F_P} \tag{7}$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \tag{8}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

A. COMPUTATIONAL EFFICIENCY

To ensure a comprehensive understanding of our proposed models' performance and efficiency, we have conducted an in-depth comparison of our achievements against existing state-of-the-art methods. Our evaluation extends beyond accuracy, precision, recall, and F1-scores to include computational efficiency, a crucial aspect for practical applications.

1) **Hardware and Optimization:** Our experiments were conducted on a MacBook M3 Max with 128GB of unified memory. This setup allowed us to benchmark the computational requirements accurately.
2) **Post Quantization on Supervised FastText:** We employed post quantization techniques to optimize the performance of our supervised FastText model. This approach was particularly beneficial for accommodating the model's scalability and efficiency without compromising accuracy. Post-quantization enabled us to adjust learning rates dynamically, with certain parameters set to true, thereby optimizing computational resource usage.
3) **ML Models Execution Time:** On average, each epoch for our ML-based models required approximately 5 minutes of execution time. This efficiency demonstrates the models' suitability for scalable applications.
4) **DL-Based Models:** The deep learning models took roughly 5 minutes per epoch, striking a balance between computational demand and performance.
5) **Transformer-Based Models:** Due to their architectural complexity, transformer-based models necessitated about 15 minutes per epoch for training. Despite the longer duration, the significant improvements in detection capabilities justify the computational investment.
6) **Model Optimization:** In addition to post-quantization, we explored various optimization techniques to enhance model efficiency further. These included layer pruning, dropout adjustments, and batch normalization, which collectively contributed to reducing overfitting and accelerating the training process.

B. ANALYSIS OF RESULTS: UNSUPERVISED FASTTEXT WITH ML AND DL MODELS

The weighted evaluation scores for ML and DL-based models, employing unsupervised FastText embeddings on WELFake, FakeNewsNet, and FakeNewsPrediction, are displayed in Tables 9, 10 and, 11 respectively. The provided tables highlight the SVM classifier's best performance across all three datasets, surpassing all other ML classifiers in both accuracy and F1-scores, achieving impressive values of 0.92, 0.97, and 0.91, respectively. Notably, it outperforms even DL-based models utilizing unsupervised FastText embeddings. This consistent and remarkable performance is noteworthy, especially considering the differing dataset sizes. The SVM classifier's ability to effectively handle high-dimensional data, create clear decision boundaries, and navigate complex, non-linear relationships makes it a strong choice for text classification, contributing to its exceptional performance in fake news detection tasks.

Unlike the SVM classifier, which demonstrated remarkable and consistent performance, ML classifiers such as LR, RF, and DT exhibited inconsistent performance across all three datasets, showing variations in their performance, even when employing different regularization techniques with unsupervised FastText embeddings. This inconsistency underscores the challenges they faced in adapting to the unique characteristics of each dataset. In contrast, all DL-based models consistently maintained their performance and generalizability across the datasets, showcasing their reliability in handling varying data complexities.

TABLE 9. Results of ML and DL-Based models with unsupervised FastText on WELFake dataset.

model	precision	recall	accuracy	f1-score
SVM	0.88	0.92	0.92	0.92
RF	0.90	0.90	0.90	0.90
LR	0.88	0.88	0.88	0.88
DT	0.83	0.83	0.83	0.83
CATBoost	0.89	0.89	0.89	0.89
XGBoost	0.88	0.88	0.88	0.88
AdaBoost	0.87	0.87	0.87	0.87
BiLSTM	0.91	0.91	0.91	0.91
BiLSTM-GRU	0.91	0.91	0.91	0.91
LSTM	0.90	0.90	0.90	0.90
CNN-LSTM	0.90	0.90	0.90	0.90
GRU	0.90	0.90	0.90	0.90
BiGRU	0.90	0.90	0.90	0.90

TABLE 10. Results of ML and DL-Based models with unsupervised FastText on FakeNewsNet dataset.

model	precision	recall	accuracy	f1-score
SVM	0.98	0.97	0.97	0.97
LR	0.94	0.94	0.94	0.94
RF	0.93	0.92	0.92	0.92
DT	0.85	0.85	0.85	0.85
CATBoost	0.95	0.95	0.95	0.95
XGBoost	0.94	0.94	0.94	0.94
AdaBoost	0.94	0.94	0.94	0.94
CNN-LSTM	0.97	0.97	0.97	0.97
BiLSTM-GRU	0.97	0.97	0.97	0.97
LSTM	0.97	0.97	0.97	0.97
BiLSTM	0.97	0.97	0.97	0.97
GRU	0.97	0.97	0.97	0.97
BiGRU	0.97	0.97	0.97	0.97

Figures 4 and 5 are the train-validation loss and accuracy curves for the CNN-LSTM model which is our best model that outperformed unsupervised FastText algorithms when stacked with the supervised FastText embeddings which will be discussed later in the details. These curves depict a stable convergence on WELFake dataset, where the validation metrics closely mirror the training metrics throughout the training process. The alignment between training and validation accuracy, coupled with a continual decrease in loss for both training and validation, suggests that the model is effectively learning and not exhibiting signs of overfitting to the training data.

TABLE 11. Results of ML and DL-Based models with unsupervised FastText on FakeNewsPrediction dataset.

model	precision	recall	accuracy	f1-score
SVM	0.91	0.91	0.91	0.91
LR	0.87	0.87	0.87	0.87
RF	0.88	0.88	0.88	0.88
DT	0.78	0.78	0.78	0.78
XGBoost	0.90	0.90	0.90	0.90
CATBoost	0.90	0.90	0.90	0.90
AdaBoost	0.88	0.88	0.88	0.88
CNN-LSTM	0.90	0.90	0.90	0.90
BiLSTM-GRU	0.90	0.90	0.90	0.90
LSTM	0.90	0.90	0.90	0.90
BiLSTM	0.90	0.90	0.90	0.90
GRU	0.90	0.90	0.90	0.90
BiGRU	0.90	0.90	0.90	0.90

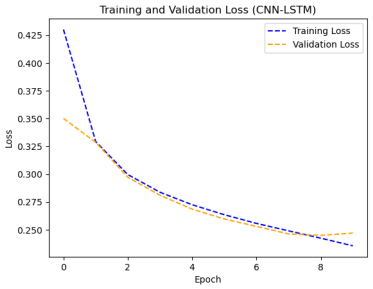

FIGURE 4. CNN-LSTM training and validation loss curve.

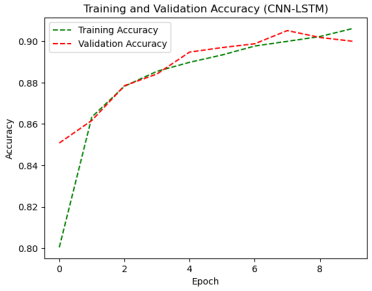

FIGURE 5. CNN-LSTM training and validation accuracy curve.

Figure 6 displays the confusion matrix for binary classification in the context of fake news detection using the CNN-LSTM model on the WELFake dataset.

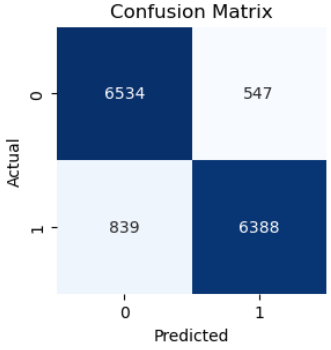

FIGURE 6. CNN-LSTM confusion matrix with unsupervised FastText.

TABLE 12. Examples of wrongly predicted instances.

Actual	Predicted	Text
0	1	a woman got fired two day working scott walker wacko trying raise fund fight
1	0	the senate right thing holding hearing replace supreme court justice scalia
1	0	apple homekit home security security smart house
0	1	the latest revelation hillary email point one thing she disqualified president
0	1	should continue support israel
1	0	barbra mocks trump at a clinton fundraiser

Table 12 highlights some examples of incorrect predictions made by the CNN-LSTM model using unsupervised FastText on the WELFake dataset.

C. ANALYSIS OF RESULTS: SUPERVISED FASTTEXT WITH ML AND DL MODELS

The weighted evaluation score for ML and DL-based models using supervised FastText embeddings have been shown in Tables 13, 14 and, 15. It can be clearly observed that in the case of supervised FastText, each ML and DL-based algorithm outperformed its counterpart in unsupervised FastText. For instance, with DT, the lowest accuracy score in unsupervised FastText was observed as 0.83, 0.85, and 0.78, respectively, across all three datasets. However, with supervised FastText, these scores showed significant enhancement, reaching 0.98, 0.91, and 0.90, respectively. In the case of SVM, which outperformed all other ML-based algorithms in unsupervised FastText, a similar consistent performance was observed with supervised FastText. However, RF surpassed all other ML-based algorithms, exhibiting accuracy scores of 0.99, 0.95, and 0.93.

Our optimal model, CNN-LSTM, surpassed all other models in ML, DL, and even those based on transformers, which will be discussed later. It achieved the highest accuracy and F1-scores of 0.99, 0.99, and 0.97, respectively, marking it as an exceptionally effective algorithm for fake news classification, even in larger datasets. The outstanding performance of CNN-LSTM can be attributed to its ability to efficiently capture both local features through CNN and long-term dependencies using LSTM, making it particularly adept at handling the complexities of natural language in fake news detection. In summary, the CNN-LSTM model not only demonstrates consistent high performance across all three datasets but also clearly outperforms or matches the performance of other ML and DL models in more complex detection scenarios. Its ability to maintain high evaluation scores of 0.99 in the second and third datasets, where ML models showed reduced effectiveness, underscores the CNN-LSTM model's advanced capability for accurately classifying fake news. This analysis, by directly referencing the specific scores from the tables, highlights the CNN-LSTM model's significant contribution to the field of fake news detection and its suitability as the proposed method in this research.

TABLE 13. Results of ML and DL-Based models with supervised FastText on WELFake dataset.

model	precision	recall	accuracy	f1-score
CNN-LSTM	0.99	0.99	0.99	0.99
BiLSTM-GRU	0.98	0.98	0.98	0.98
LSTM	0.98	0.98	0.98	0.98
BiLSTM	0.98	0.98	0.98	0.98
GRU	0.98	0.98	0.98	0.98
BiGRU	0.98	0.98	0.98	0.98
SVM	0.99	0.99	0.99	0.99
RF	0.99	0.99	0.99	0.99
LR	0.99	0.99	0.99	0.99
DT	0.98	0.98	0.98	0.98
CATBoost	0.99	0.99	0.99	0.99
XGBoost	0.99	0.99	0.99	0.99
AdaBoost	0.99	0.99	0.99	0.99

TABLE 14. Results of ML and DL-Based models with supervised FastText on FakeNewsNet dataset.

model	precision	recall	accuracy	f1-score
CNN-LSTM	0.99	0.99	0.99	0.99
BiLSTM-GRU	0.97	0.97	0.97	0.97
LSTM	0.99	0.99	0.99	0.99
BiLSTM	0.99	0.99	0.99	0.99
GRU	0.99	0.99	0.99	0.99
BiGRU	0.99	0.99	0.99	0.99
CATBoost	0.95	0.95	0.95	0.95
XGBoost	0.94	0.94	0.94	0.94
AdaBoost	0.94	0.94	0.94	0.94
RF	0.95	0.95	0.95	0.95
SVM	0.94	0.94	0.94	0.94
LR	0.93	0.93	0.93	0.93
DT	0.91	0.91	0.91	0.91

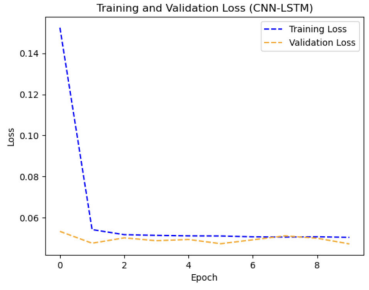

FIGURE 7. Supervised CNN-LSTM training and validation loss curve.

Figures 7 and 8 indicate a stable convergence, with the validation metrics closely tracking the training metrics across epochs. The graphs demonstrate a stable convergence and

TABLE 15. Results of ML and DL-Based models with supervised FastText on FakeNewsPrediction dataset.

model	precision	recall	accuracy	f1-score
CNN-LSTM	0.97	0.97	0.97	0.97
BiLSTM-GRU	0.96	0.96	0.96	0.96
LSTM	0.97	0.97	0.97	0.97
BiLSTM	0.96	0.96	0.96	0.96
GRU	0.97	0.97	0.97	0.97
BiGRU	0.97	0.97	0.97	0.97
CATBoost	0.92	0.92	0.92	0.92
XGBoost	0.93	0.93	0.93	0.93
AdaBoost	0.92	0.92	0.92	0.92
RF	0.93	0.93	0.93	0.93
SVM	0.92	0.92	0.92	0.92
LR	0.91	0.91	0.91	0.91
DT	0.90	0.90	0.90	0.90

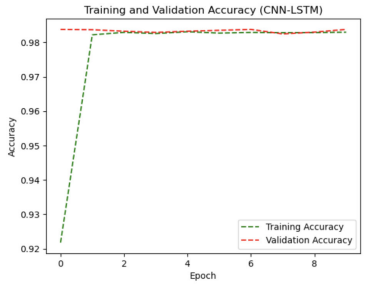

FIGURE 8. Supervised CNN-LSTM training and validation accuracy curve.

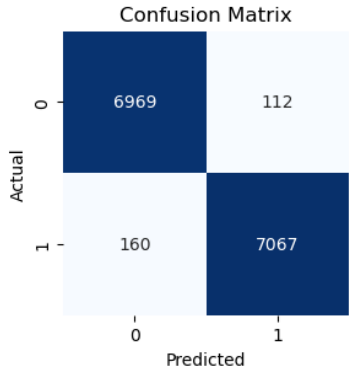

FIGURE 9. CNN-LSTM confusion matrix with supervised FastText.

indicate that the validation scores are in close agreement with the training scores throughout the training epochs. The close alignment between training and validation accuracy, alongside a consistent decrease in loss for both training and validation, suggests that the model is learning generalizable patterns rather than overfitting the training data.

Figure 9 represents the confusion matrix for CNN-LSTM using supervised FastText.

D. ANALYSIS OF RESULTS: TRANSFORMER BASED MODELS

The following figures 16, 17, and 18 represents the results obtained from transformer-based models which are BERT with its base and large variant, XLNet and RoBERTa.

TABLE 16. Results of transformer based models on WELFake dataset.

model	precision	recall	accuracy	f1-score
$BERT_{base}$	0.97	0.97	0.97	0.97
$BERT_{large}$	0.97	0.97	0.97	0.97
$XLNet_{base}$	0.97	0.97	0.97	0.97
$RoBERTa_{large}$	0.96	0.96	0.96	0.96

TABLE 17. Results of transformer based models on FakeNewsNet dataset.

model	precision	recall	accuracy	f1-score
$BERT_{base}$	0.99	0.99	0.99	0.99
$BERT_{large}$	0.99	0.99	0.99	0.99
$XLNet_{base}$	0.99	0.99	0.99	0.99
$RoBERTa_{large}$	0.99	0.99	0.99	0.99

TABLE 18. Results of transformer based models on FakeNewsPrediction dataset.

model	precision	recall	accuracy	f1-score
$BERT_{base}$	0.95	0.95	0.95	0.95
$BERT_{large}$	0.95	0.95	0.95	0.95
$XLNet_{base}$	0.97	0.97	0.97	0.97
$RoBERTa_{large}$	0.95	0.95	0.95	0.95

On the WELFake Dataset, all models, including BERT base, BERT large, XLNet base, and RoBERTa large, show a strong and uniform performance, with each scoring 0.97 in precision, recall, accuracy, and F1-score, except RoBERTa large which is marginally lower at 0.96. This slight underperformance of RoBERTa large might indicate some dataset-specific challenges or limitations in model architecture. Moving to the FakeNewsNet Dataset, a remarkable increase in model performance is observed, with all models achieving a uniform score of 0.99 across all metrics. This exceptional performance suggests that the FakeNewsNet Dataset contains patterns more easily interpreted by these models compared to the other datasets. In the case of the FakeNewsPrediction Dataset, a slight variation is noted. While BERT base, BERT large, and RoBERTa large models maintain a consistent score of 0.95 across all metrics, the XLNet base model demonstrates superior performance with scores of 0.97.

Despite the consistent and robust performance of the transformer-based models, the CNN-LSTM architecture exceeded the effectiveness of all other learning algorithms used in this research. This outcome highlights the CNN-LSTM model's superior capability in handling the specific challenges and nuances of the datasets used.

V. COMPARISON OF THE RESULTS WITH THE STATE-OF-THE-ART

In this section, we compare our results with those from two baseline studies, [29] and [30]. In both the WELFake and FakeNewsNet datasets, the proposed models achieved a remarkable accuracy of 0.99, surpassing the baseline scores of 0.97. This improvement underscores the effectiveness of the applied methodologies in our study. While the

0.02 increase in accuracy may appear marginal at first glance, it is statistically significant when considering the extensive size of the datasets involved. Specifically, the WELFake dataset includes 72,134 records, and the FakeNewsNet dataset contains 23,196 records. By implementing strategic regularization techniques and meticulous parametric tuning, we were able to achieve these promising results. Such approaches not only enhance model performance but also contribute to the robustness and generalizability of the models. This suggests that our models are not only adept at handling the specific datasets they were trained on but also have the potential to perform well across varied datasets, demonstrating their adaptability and reliability in broader applications.

TABLE 19. Comparison of the results with SOTA.

Baseline Results	
Dataset	Accuracy Score
WELFake[29]	0.97
FakeNewsNet[30]	0.97
Proposed Work	
Dataset	Accuracy Score
WELFake	0.99
FakeNewsNet	0.99

VI. INTERPRETABILITY MODELING

Interpretability modeling is crucial in the field of fake news detection, as it helps create models or methods that make complex machine learning processes more clear and transparent. Techniques like LDA and LIME are particularly useful in this area. LDA helps uncover hidden themes in large text datasets, which is vital for understanding the content patterns that might indicate fake news. On the other hand, LIME provides straightforward explanations for individual predictions made by the models, shedding light on the reasons behind classifying certain news articles as fake.

A. TOPIC MODELING WITH LDA

LDA is renowned for its effective balance between simplicity and sophistication in topic modeling. It is crafted to identify various topics within a corpus of text. These identified topics can be understood as groups of words that often appear together [57]. In our research, we assessed the performance of LDA using two key metrics: coherence and perplexity. Coherence evaluates how meaningful and interpretable the topics generated by LDA are, by assessing the semantic similarity between words within these topics. Perplexity, on the other hand, measures how well the model predicts a sample. The calculations for coherence and perplexity are detailed in equations 10 and 11, respectively, as cited in [58] and [59].

$$Coherence = \frac{1}{C} * \sum PMI(w_i, w_j) \tag{10}$$

$$Perplexity = exp\left[-1 * \frac{log\ \ likelihood}{total\ \ number\ \ of\ \ words}\right] \tag{11}$$

We applied LDA to WELFake, FakeNewsNet, and FakeNewsPrediction datasets mentioned in this paper. Based on the coherent terms identified, we categorized each dataset into three primary topics, providing a structured thematic understanding of the datasets. The hyperparameter tuning of LDA is performed by performing different experiments and the best parameters obtained, which are used in this study are shown in Table 20.

TABLE 20. Hyperparameter tuning of LDA model.

Parameters	LDA Model
No. of Topics	3
Random State	100
Max Features	1000
Update Every	1
Chunk Size	100
Alpha	Auto
Stop Words	'English'

Table 22 shows LDA results across the three datasets indicate a solid performance in terms of both coherence and perplexity scores, suggesting a good understanding and effective topic modeling of the datasets. For the WELFake dataset, the coherence score is 0.26236, coupled with a perplexity score of -8.218491. These figures imply that the topics generated are reasonably coherent and meaningful, with the negative perplexity indicating a good fit of the model to the data. In the case of the FakeNewsPrediction dataset, the coherence score is slightly lower at 0.22106, yet still represents a decent level of topic interpretability and relevance. The perplexity score is very close to that of WELFake, at -8.2188, indicating a consistent model performance across different datasets. Finally, the FakeNewsNet dataset shows a coherence score of 0.26039 and a perplexity of -8.9435. The coherence is comparable to that of WELFake, suggesting effective topic representation. The relatively lower perplexity score here signifies an even better model fit to the dataset compared to the other two. Overall, these scores reflect that LDA has a good grasp on the datasets, effectively capturing the underlying thematic structures in the text. This demonstrates the utility of LDA in providing meaningful insights into large text corpora, particularly in the context of fake news detection and analysis.

B. LIME

Local Interpretable Model-Agnostic Explanations (LIME) is a method designed for providing localized insights and evaluating the predictions of any learning algorithm. It offers an understanding of how a model's predictions correspond to the specific requirements of a given task. This technique is especially useful in situations where comprehending a model's decision-making process is as crucial as the accuracy of its outcomes, as noted by [60]. The LIME formula seeks to identify an interpretable model, denoted as $\hat{g}$, within a class of models G. It aims to minimize the loss $\mathcal{L}$, which

TABLE 21. Topics and items for four datasets.

Dataset	Topic	Terms
WELFake	U.S. Presidential Politics	trump, clinton, hillary, president, donald, people, time
	Election Campaigns and Law Enforcement	police, campaign, trump, republican, clinton, states
	National and International Governance	government, president, states, united, reuters, obama
FakeNewsPrediction	U.S. Political Campaigns	'trump', 'clinton', 'campaign', 'republican', 'hillary', 'election', 'sanders', 'donald'
	U.S. Presidency and Governance	'obama', 'president', 'people', 'government', 'states', 'house', 'years', 'american'
	International Relations and Security	'state', 'police', 'clinton', 'russia', 'war', 'world', 'time'
FakeNewsNet	Celebrity Lifestyle and Entertainment	'kardashian', 'jenner', 'kim', 'selena', 'gomez', 'kylie', 'justin', 'awards', 'baby'
	Celebrity Relationships and Weddings	'jennifer', 'meghan', 'prince', 'markle', 'brad', 'harry', 'pitt', 'kate', 'angelina', 'wedding
	Entertainment Industry and Media	'new', 'taylor', 'swift', 'watch', 'tv', 'video', 'news', 'season', 'blake',

TABLE 22. LDA results across all datasets.

Dataset	Coherence	Perplexity
WELFake	0.2623	-8.2184
FakeNewsPrediction	0.2210	-8.2188
FakeNewsNet	0.2603	-8.9435

measures the discrepancy between the predictions of g and the more complex model f. This is done while accounting for the locality kernel π_x. Additionally, $\Omega(g)$ represents the complexity of the interpretable model g, with a preference for lower complexity to ensure better interpretability and maintain simplicity.

$$\hat{g} = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{12}$$

LIME is a widely-used interpretability technique known for its simplicity and effectiveness in providing insights into complex machine learning models. Other techniques like SHAP, counterfactual explanations, and similar language tools can help understand complex models. But we chose LIME because it uses simpler methods, making explanations easy to understand. LIME's use of Lasso or short trees helps create straightforward and focused explanations, which are easier for humans to understand [61]. While other techniques offer valuable interpretability features, we opted for LIME due to its ability to provide local explanations, which are crucial for understanding individual predictions within our model. LIME generates locally faithful explanations by approximating the decision boundary around a specific instance, thus providing insights into why a model made a particular prediction for that instance. This approach aligns well with our goal of gaining deeper insights into the model's decision-making process, particularly in the context of hate speech detection in multimedia-rich content.

In this study we will implement LIME with supervised FastText CNN-LSTM which showed the best performance across all three datasets, we will use some examples from the WELFake dataset to interpret the model's decision-making process. The LIME visualization provided in figure 10 offers a granular view into the decision-making process of the learning algorithm. According to the prediction probabilities, the learning algorithm has confidently classified this text as fake news (label 0) with a probability of 1.00, and there is no probability assigned to it being real news (label 1). The right side of the visualization highlights key words that

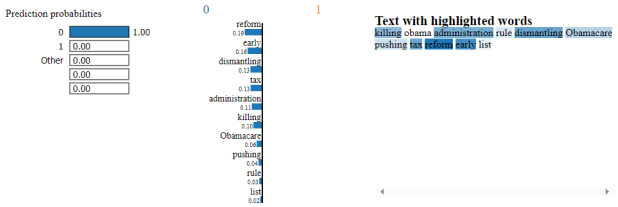

FIGURE 10. Example 1: Supervised CNN-LSTM with LIME.

have likely contributed to the model's prediction, with words like "reform", "early", "dismantling", "tax", "administration", "killing", "Obamacare", "pushing", "rule", and "list". These words are weighted according to their influence on the prediction, as indicated by the numbers next to each word. The terms such as "Obamacare", "reform", and "dismantling" have higher weights, suggesting they are strong indicators for the model's decision to classify this as fake news.

In figure 11 LIME visualization depicts the model's prediction process for another news article, which indicates that the learning algorithm has assigned a higher probability to the text being fake news (label 0) with a probability of 0.80, while there is a smaller probability of 0.20 for the text being real news (label 1). The words highlighted on the right side as influencing the model's decision include "disqualified", "president", "thing", "latest", "revelation", "email", "one", "Hillary", and "point". The weights next to the words suggest their relative impact on the prediction, with "disqualified" and "president" having the most significant influence. The model appears to be using these key terms to assess the credibility of the text, with words related to political figures and potential controversies ("Hillary", "email") being central to its classification as fake news. The presence of words like "latest" and "revelation" might indicate a news-like structure which could contribute to the ambiguity in the model's decision, reflected in the less confident prediction compared to the first example.

In figure 12, LIME visualization shows that the learning algorithm is completely certain in its classification, assigning a probability of 1.00 to the text being fake news (label 0) and no probability of it being real (label 1).

The visualization highlights the terms "Reuters", "EDIT", "Source", "Twitter", "realDonaldTrump", "confirmed", "edited", and "posted" as key contributors to the prediction. Notably, the word "Reuters" carries the highest weight, followed by "EDIT" and "Source," which

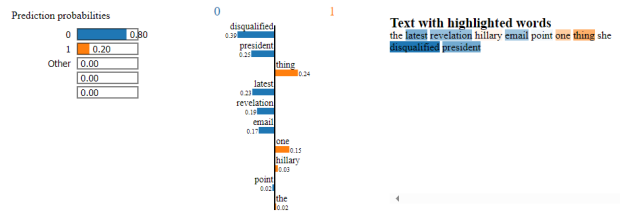

FIGURE 11. Example 2: Supervised CNN-LSTM with LIME.

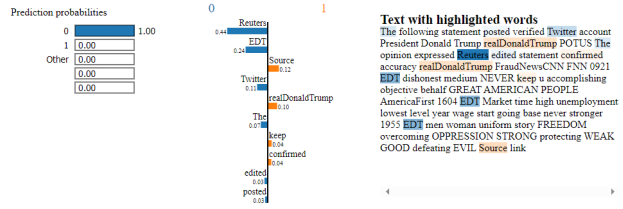

FIGURE 12. Example 3: Supervised CNN-LSTM with LIME.

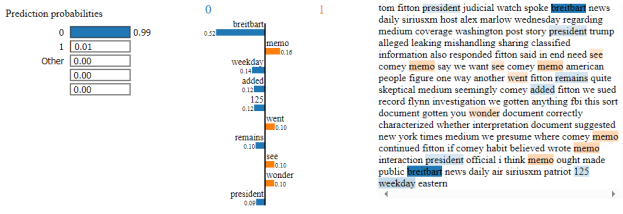

FIGURE 13. Example 4: Supervised CNN-LSTM with LIME.

may indicate that the presence of these terms is strongly associated with the features of fake news within the model's learned parameters. Additionally, the inclusion of "realDonaldTrump" and "FraudNewsCNN" could suggest the model is picking up on the mention of high-profile names and possible claims of misinformation as indicators of fake news.

In figure 13, learning algorithm's prediction with a high degree of confidence, indicating a 0.99 probability that the text is fake news (label 0) and a very small probability of 0.01 for it being real news (label 1).

Key words influencing this decision include "breitbart", "memo", "weekday", "added", "went", "remains", "see", "wonder", and "president". The term "breitbart" has the highest weight, which suggests that the model has learned to associate this term with the characteristics of fake news. Words like "memo" and temporal references such as "weekday" and "12:25" also seem to play a significant role in the classification.

LIME visualization in figure 14 indicates that the learning algorithm misclassified a fake news article (label 0) as real news (label 1) with a confidence of 0.71. The LIME output highlights the words "October", "said", "November", "massive", "Britain", "French", "France", and "March" as significant for the model's decision-making process. The weightiest term, "October" along with other time-related terms like "November" and "March", and frequent occurrences of the word "said," might have misled the model into

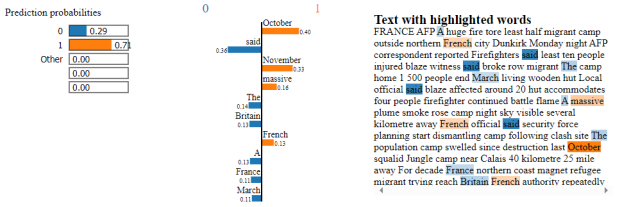

FIGURE 14. Example 5: Supervised CNN-LSTM with LIME.

perceiving the text as a legitimate news report. This could indicate that the model may be overvaluing specific keywords that are often found in factual news content, without fully grasping the deceptive context that characterizes fake news.

VII. CONCLUSION AND FUTURE WORK

This research study represents a comprehensive and robust framework for the detection of fake news, addressing the pressing challenge of misinformation spread across social media. By utilizing three diverse datasets, we systematically evaluated the efficacy of FastText embeddings in conjunction with advanced ML and DL-based techniques, with an emphasis on optimization strategies to avoid overfitting and address generalizability. Our findings indicate that a hybrid model combining CNN with LSTM layers, augmented with FastText embeddings, outperforms other models in accurately classifying news articles. Moreover, the application of transformer-based models has shed light on the capabilities of these architectures in deciphering complex syntactic structures for enhanced semantic understanding. The use of explainable AI through LIME and LDA has not only justified the transparency of the detection process but also provided valuable interpretative insights. In future work, we will aim to expand the scope of fake news detection to encompass multiple languages, particularly those with scarce linguistic resources. We aim to explore the potential of multilingual transformers, such as mBERT, mT5, and GPT, to tackle the intricacies of multi-label classification, thereby advancing the frontier in combating the global challenge of fake news dissemination. Furthermore, by adopting adversarial training methods, we anticipate further advancements in our fight against the widespread issue of fake news dissemination globally.

ACKNOWLEDGMENT

This research work has been acknowledged by SOCYTI.[6]

REFERENCES

[1] R. M. Johnson, "Social media and free speech: A collision course that threatens democracy," *Ohio Northern Univ. Law Rev.*, vol. 49, no. 2, p. 5, 2023.
[2] S. Rastogi and D. Bansal, "A review on fake news detection 3T's: Typology, time of detection, taxonomies," *Int. J. Inf. Secur.*, vol. 22, no. 1, pp. 177–212, Feb. 2023.
[3] S. Rastogi and D. Bansal, "Disinformation detection on social media: An integrated approach," *Multimedia Tools Appl.*, vol. 81, no. 28, pp. 40675–40707, Nov. 2022.

[6]https://www.bigdata.vestforsk.no/ongoing/socyti

[4] N. Capuano, G. Fenza, V. Loia, and F. D. Nota, "Content-based fake news detection with machine and deep learning: A systematic review," *Neurocomputing*, vol. 530, pp. 91–103, Apr. 2023.

[5] F. Miró-Llinares and J. C. Aguerri, "Misinformation about fake news: A systematic critical review of empirical studies on the phenomenon and its status as a'threat,"' *Eur. J. Criminol.*, vol. 20, no. 1, pp. 356–374, Jan. 2023.

[6] C. Silverman, "This analysis shows how viral fake election news stories outperformed real news on Facebook," *BuzzFeed news*, vol. 16, p. 24, Jan. 2016.

[7] G. Sansonetti, F. Gasparetti, G. D'Aniello, and A. Micarelli, "Unreliable users detection in social media: Deep learning techniques for automatic detection," *IEEE Access*, vol. 8, pp. 213154–213167, 2020.

[8] A. Jarrahi and L. Safari, "Evaluating the effectiveness of publishers' features in fake news detection on social media," *Multimedia Tools Appl.*, vol. 82, no. 2, pp. 2913–2939, Jan. 2023.

[9] R. Rodríguez-Ferrándiz, "An overview of the fake news phenomenon: From untruth-driven to post-truth-driven approaches," *Media Commun.*, vol. 11, no. 2, pp. 15–29, Apr. 2023.

[10] M. R. Kondamudi, S. R. Sahoo, L. Chouhan, and N. Yadav, "A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 6, Jun. 2023, Art. no. 101571.

[11] C. Martel and D. G. Rand, "Misinformation warning labels are widely effective: A review of warning effects and their moderating features," *Current Opinion Psychol.*, vol. 54, Dec. 2023, Art. no. 101710.

[12] S. Wang, "Factors related to user perceptions of artificial intelligence (AI)-based content moderation on social media," *Comput. Hum. Behav.*, vol. 149, Dec. 2023, Art. no. 107971.

[13] K. Węcel, M. Sawiński, M. Stróżyna, W. Lewoniewski, E. Księżniak, P. Stolarski, and W. Abramowicz, "Artificial intelligence—Friend or foe in fake news campaigns," *Econ. Bus. Rev.*, vol. 9, no. 2, pp. 41–70, 2023.

[14] A. Altheneyan and A. Alhadlaq, "Big data ML-based fake news detection using distributed learning," *IEEE Access*, vol. 11, pp. 29447–29463, 2023.

[15] S. D. M. Kumar and A. M. Chacko, "A systematic survey on explainable AI applied to fake news detection," *Eng. Appl. Artif. Intell.*, vol. 122, Jun. 2023, Art. no. 106087.

[16] S. Ali, F. Akhlaq, A. S. Imran, Z. Kastrati, S. M. Daudpota, and M. Moosa, "The enlightening role of explainable artificial intelligence in medical & healthcare domains: A systematic literature review," *Comput. Biol. Med.*, vol. 166, Nov. 2023, Art. no. 107555.

[17] D. Choudhury and T. Acharjee, "A novel approach to fake news detection in social networks using genetic algorithm applying machine learning classifiers," *Multimedia Tools Appl.*, vol. 82, no. 6, pp. 9029–9045, Mar. 2023.

[18] W. Wang, "A new benchmark dataset for fake news detection," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2021.

[19] S. Dutta and S. K. Bandyopadhyay, "Fake job recruitment detection using machine learning approach," *Int. J. Eng. Trends Technol.*, vol. 68, no. 4, pp. 48–53, Apr. 2020.

[20] L. R. Ali, B. N. Shaker, and S. A. Jebur, "An extensive study of sentiment analysis techniques: A survey," in *AIP Conf. Proc.*, 2023.

[21] M. A. Chandra and S. S. Bedi, "Survey on SVM and their application in image classification," *Int. J. Inf. Technol.*, vol. 13, no. 5, pp. 1–11, Oct. 2021.

[22] H. Wang, F. G. Quintana, Y. Lu, M. Mohebujjaman, and K. Kamronnaher, "An application of ordianl logistic regression model to a health survey in a hispanic university," Tech. Rep.

[23] J. Hu and S. Szymczak, "A review on longitudinal data analysis with random forest in precision medicine," 2022, *arXiv:2208.04112*.

[24] M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," *IEEE Netw.*, vol. 30, no. 3, pp. 22–29, May 2016.

[25] S. Lee, J. Lee, H. Moon, C. Park, J. Seo, S. Eo, S. Koo, and H. Lim, "A survey on evaluation metrics for machine translation," *Mathematics*, vol. 11, no. 4, p. 1006, Feb. 2023.

[26] V. Bhaskar and U. Shanmugam, "Novel spam comment detection system using countvectorizer techniques with SVM for online YouTube comments for improving the recall and precision value over naive Bayes," in *Proc. AIP Conf.*, 2023.

[27] P. Akhtar, A. M. Ghouri, H. U. R. Khan, M. Amin ul Haq, U. Awan, N. Zahoor, Z. Khan, and A. Ashraf, "Detecting fake news and disinformation using artificial intelligence and machine learning to avoid supply chain disruptions," *Ann. Operations Res.*, vol. 327, no. 2, pp. 633–657, Aug. 2023.

[28] A. K. Shalini, S. Saxena, and B. S. Kumar, "Designing a model for fake news detection in social media using machine learning techniques," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 2, pp. 218–226, 2023.

[29] P. K. Verma, P. Agrawal, I. Amorim, and R. Prodan, "WELFake: Word embedding over linguistic features for fake news detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 881–893, Aug. 2021.

[30] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big Data*, vol. 8, no. 3, pp. 171–188, Jun. 2020.

[31] C.-O. Truică and E.-S. Apostol, "It's all in the embedding! Fake news detection using document embeddings," *Mathematics*, vol. 11, no. 3, p. 508, Jan. 2023.

[32] J. H. Joloudari, S. Hussain, M. A. Nematollahi, R. Bagheri, F. Fazl, R. Alizadehsani, R. Lashgari, and A. Talukder, "BERT-deep CNN: State of the art for sentiment analysis of COVID-19 tweets," *Social Netw. Anal. Mining*, vol. 13, no. 1, p. 99, Jul. 2023.

[33] D. Antony, S. Abhishek, S. Singh, S. Kodagali, N. Darapaneni, M. Rao, and A. R. Paduri, "A survey of advanced methods for efficient text summarization," in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Mar. 2023, pp. 0962–0968.

[34] J. Briskilal and C. N. Subalalitha, "An ensemble model for classifying idioms and literal texts using BERT and RoBERTa," *Inf. Process. Manage.*, vol. 59, no. 1, Jan. 2022, Art. no. 102756.

[35] S. J. Johnson, M. R. Murty, and I. Navakanth, "A detailed review on word embedding techniques with emphasis on word2vec," *Multimedia Tools Appl.*, vol. 2023, pp. 1–29, Oct. 2023.

[36] M. Umer, Z. Imtiaz, M. Ahmad, M. Nappi, C. Medaglia, G. S. Choi, and A. Mehmood, "Impact of convolutional neural network and FastText embedding on text classification," *Multimedia Tools Appl.*, vol. 82, no. 4, pp. 5569–5585, Feb. 2023.

[37] A. Nanade and A. Kumar, "Combating fake news on Twitter: A machine learning approach for detection and classification of fake tweets," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 1, pp. 424–436, 2024.

[38] P. K. Verma, P. Agrawal, V. Madaan, and R. Prodan, "MCred: Multi-modal message credibility for fake news detection using BERT and CNN," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 8, pp. 10617–10629, Aug. 2023.

[39] Z. Guo, Q. Zhang, F. Ding, X. Zhu, and K. Yu, "A novel fake news detection model for context of mixed languages through multiscale transformer," *IEEE Trans. Computat. Social Syst.*, 2024.

[40] A. Praseed, J. Rodrigues, and P. S. Thilagam, "Hindi fake news detection using transformer ensembles," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105731.

[41] S. Sai, A. W. Jacob, S. Kalra, and Y. Sharma, "Stacked embeddings and multiple fine-tuned XLM-roBERTa models for enhanced hostility identification," in *Combating Online Hostile Posts in Regional Languages During Emergency Situation*. Cham, Switzerland: Springer, 2021, pp. 224–235.

[42] K. Subramanyam Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS : A survey of transformer-based pretrained models in natural language processing," 2021, *arXiv:2108.05542*.

[43] M. Bhardwaj, M. Shad Akhtar, A. Ekbal, A. Das, and T. Chakraborty, "Hostility detection dataset in Hindi," 2020, *arXiv:2011.03588*.

[44] S. Biradar, S. Saumya, and A. Chauhan, "Combating the infodemic: COVID-19 induced fake news recognition in social media networks," *Complex Intell. Syst.*, vol. 9, no. 3, pp. 2879–2891, Jun. 2023.

[45] M. S. I. Malik, A. Nawaz, M. M. Jamjoom, and D. I. Ignatov, "Effectiveness of ELMo embeddings, and semantic models in predicting review helpfulness," *Intell. Data Anal.*, vol. 2023, pp. 1–21, Nov. 2023.

[46] J. Wu, W. Xu, Q. Liu, S. Wu, and L. Wang, "Adversarial contrastive learning for evidence-aware fake news detection with graph neural networks," *IEEE Trans. Knowl. Data Eng.*, 2023.

[47] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the Web and social media," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 1003–1012.

[48] A. Vlachos and S. Riedel, "Fact checking: Task definition and dataset construction," in *Proc. ACL Workshop Lang. Technol. Comput. Social Sci.*, 2014, pp. 18–22.
[49] K. Soga, S. Yoshida, and M. Muneyasu, "Exploiting stance similarity and graph neural networks for fake news detection," *Pattern Recognit. Lett.*, vol. 177, pp. 26–32, Jan. 2024.
[50] I. A. Pilkevych, D. L. Fedorchuk, M. P. Romanchuk, and O. M. Naumchak, "An analysis of approach to the fake news assessment based on the graph neural networks," in *Proc. CEUR Workshop*, vol. 3374, 2023, pp. 56–65.
[51] T. J. Billard and R. E. Moran, "Designing trust: Design style, political ideology, and trust in 'fake' news websites," *Digit. Journalism*, vol. 11, no. 3, pp. 519–546, Mar. 2023.
[52] P. P. Ray, "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope," *Internet Things Cyber-Phys. Syst.*, vol. 3, pp. 121–154, Jan. 2023.
[53] R. S. Satpute and A. Agrawal, "A critical study of pragmatic ambiguity detection in natural language requirements," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 3s, pp. 249–259, 2023.
[54] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," 2017, *arXiv:1712.09405*.
[55] C. Qiao, B. Huang, G. Niu, D. Li, D. Dong, W. He, D. Yu, and H. Wu, "A new method of region embedding for text classification," in *Proc. ICLR*, 2018.
[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023.
[57] S. N. Edi, "Topic modelling Twitter data with latent Dirichlet allocation method," Tech. Rep., 2022.
[58] D. M. Mimno, H. M. Wallach, E. M. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 262–272.
[59] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno, "Evaluation methods for topic models," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 1105–1112.
[60] P. Biecek and T. Burzykowski, "Local interpretable model-agnostic explanations (LIME)," *Explanatory Model Anal. Explore, Explain Examine Predictive Models*, vol. 1, pp. 107–124, Jan. 2021.
[61] H. Mehta and K. Passi, "Social media hate speech detection using explainable artificial intelligence (XAI)," *Algorithms*, vol. 15, no. 8, p. 291, Aug. 2022.

EHTESHAM HASHMI received the B.S. degree in computer science from the University of Central Punjab, Lahore Campus, in 2020, and the M.S. degree in computer science from COMSATS University Islamabad, Lahore Campus, in 2022. He is currently pursuing the Ph.D. degree with the Department of Information Security and Communication Technology (IIK), Norwegian University of Science and Technology (NTNU). From 2022 to 2023, he was a Lecturer with the Department of Computer Science, The University of Lahore. His research interests include multilingual natural language processing, computational linguistics, large language models, knowledge graphs, and data mining.

SULE YILDIRIM YAYILGAN received the M.Sc. degree in computer engineering, in 1995, and the Ph.D. degree in artificial intelligence and computer science, in 2002. She has been with the Department of Information Security and Communication Technology (IIK), NTNU, since 2009. She worked for more than 25 years in academic teaching. She has been supervising students at different academic levels and has been publishing more than 100 journals and conferences papers. Her research interests include image processing, information security, natural language processing, computational linguistics, large language models, and data mining.

MUHAMMAD MUDASSAR YAMIN is currently an Associate Professor with the Department of Information and Communication Technology, Norwegian University of Science and Technology (NTNU). He is a member with the System Security Research Group, and the focus of his research is on system security, penetration testing, security assessment, and intrusion detection. Before joining NTNU, he was an Information Security Consultant and served multiple government and private clients. He holds multiple cybersecurity certifications, such as OSCE, OSCP, LPT-MASTER, CEH, CHFI, CPTE, CISSO, and CBP.

SUBHAN ALI received the bachelor's degree from Sukkur IBA University, through a fully funded Talent Hunt Scholarship offered by OGDCL, Pakistan, in 2021. He is currently pursuing the master's degree in applied computer science with Norwegian University of Science and Technology (NTNU), Norway, through a NORPART-CONNECT fully funded scholarship. He is a highly motivated Researcher with a passion for advancing the field of artificial intelligence. His research interests include the intersection of explainable AI, generative AI, and natural language processing. His talent for innovative problem-solving and his dedication to advancing the field of AI makes him a valuable addition to any team.

MOHAMED ABOMHARA received the bachelor's degree in Libya, in 2006, the master's degree (M.Sc.) in computer science in Malaysia, in 2011, the master's degree (M.B.A.) in business administration, in 2013, and the Ph.D. degree in information technology from the University of Agder, Norway, in 2018. He is currently a Cybersecurity Researcher and a Data Protection Specialist with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU). His scholarly contributions extend to active participation in several prestigious European, Erasmus+, and Norwegian Research Council projects, where he has assumed both technical and managerial roles and has published multiple journals and conferences papers. His commitment to advancing technology while upholding ethical and privacy standards underscores his prominent role in academia and research. His research interests include the development of data-driven technologies that uphold critical principles, such as transparency, accountability, and privacy.