Journal of Computer Science

Research Article

Fake News Detection Using Weighted Fine-Tuned BERT and Sparse Recurrent Neural Network

Asha Kathigi, Mukta Pujar, A M S Akshatha, RN Shilpa and Shivaranjani S. Shirabadagi

Department of Artificial Intelligence and Machine Learning, GM University, Davanagere, India

Article history
Received: 26-12-2024
Revised: 03-06-2025
Accepted: 25-07 2025

Corresponding Author:
Asha Kathigi
Department of Artificial intelligence and Machine Learning, GM University, Davanagere, India
Email: ashak@gmit.ac.in

Abstract: Fake news refers to misinformation or false reports shared in the form of images, articles, or videos, disguised as real news to manipulate people's opinions. Recently, fake news and rumors have spread extensively and rapidly around the world. This has led to the production and propagation of inaccurate news articles. Therefore, it is necessary to restrict the spread of fake information in the media to establish confidence globally. For this purpose, this research proposes Weighted Fine-tuned-Bidirectional Encoder Representations from Transformers-based Sparse Recurrent Neural Network (WFT-BERT-SRNN) for fake news detection through Deep Learning (DL). Data preprocessing is established using stop word removal, tokenization, and stemming to eliminate unwanted phrases or words. Then, WFT-BERT is employed for feature extraction, and finally, SRNN is employed to detect and classify fake news as real or fake. WFT-BERT-SRNN achieves a superior accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 on the BuzzFeed, PolitiFact, Fakeddit, and Weibo datasets compared to existing techniques like DeepFake and image caption-based technique.

Keywords: Deep Learning, Fake News, Natural Language Processing, Sparse Recurrent Neural Network and Weighted Fine-Tuned-Bidirectional Representation for Transformers

Introduction

Fake news is one of the primary sources of danger to democracy, journalism, and global commerce, with significant collateral harm (Saleh *et al*., 2021). Social networking platforms serve as a major channel through which consumers build, access, and share diverse information. The deployment of social media networks has grown as various users receive and search for the latest updates at relevant times. Moreover, social media provides an opportunity for rapidly spreading numerous fake and misleading data to users, which can have destructive consequences for communities (Dixit *et al*., 2022). Fake news spreads faster and more widely than factual news, causing significant societal and individual harm (Liao *et al*., 2022). Facebook is one of the most commonly used media platforms for spreading fake news, surpassing Twitter, Google, and webmail services like Gmail and Yahoo (Elsaeed *et al*., 2021). The concerns surrounding fake news have only intensified with the increasing time people spend on social media, making it the primary source of news for many (Rai *et al*., 2022). Additionally, certain aggregators of official news intentionally spread false information to gain popularity, generate revenue, or achieve political objectives. This ease of dissemination and lack of control over the internet enable fake news to spread widely (Aslam *et al*., 2021).

During the 2016 US election, numerous fake news instances were reported to have been spread on social media platforms, including a false claim about a new Air Marshal nomination in India during the presidential elections (Altheneyan and Alhadlaq, 2023). The effectiveness of evaluating and sharing knowledge with others enables engaging online social networks. Moreover, the rapid distribution of instantaneous data with minimal effort facilitates the widespread diffusion of false data and news (Jain *et al*., 2023). Disinformation and misinformation are the two kinds of fake information. Disinformation refers to false data deliberately disseminated to mislead the public and is intended to cause political, economic, and social impacts (Kumar *et al*., 2023). Furthermore, it is challenging to prevent fake news dissemination as it is repeatedly shared on a large scale (Seddari *et al*., 2022). Thus, the spread of fake news has exerted a negative impact on social, personal, and political dynamics (Nasir *et al*., 2021). Hence, fake news detection has become a critical necessity on social media (Wu *et al*., 2021). To address this issue, an efficient detection technique is proposed in this study to combat malicious intentions and help users avoid falling for fake news (Rezaei *et al*., 2022; Ozbay and Alatas, 2021). Recent research models based on fake news detection using DL have attained impressive success by employing various features of social media news such as user features, text data, and user feedback (Albahar, 2021). Fake news and rumors continue to spread extensively and rapidly around the world, leading to the production and propagation of inaccurate news articles. Therefore, it is essential to restrict the spread of fake information in the media to restore global trust. In order to address this challenge, WFT-BERT-SRNN is proposed for fake news detection, integrating the contextual understanding of BERT with the efficiency of SRNN. BERT captures the deep semantic meaning of news content, while SRNN focuses on relevant features with lower computational complexity. The weighted fine-tuning method enhances the model's performance in detecting deceptive patterns, ensuring accurate fake news detection.

This research introduces a novel approach to fake news detection, centered on a hybrid architecture that synergizes deep contextual language understanding with structured sequential analysis. The primary contribution is WFT-BERT (Weighted Fine-Tuned BERT), a feature extraction module fine-tuned with a specialized weighted loss to compel the model to concentrate on deceptive linguistic patterns. This deep bidirectional learning captures rich semantic and contextual nuances, enabling the identification of subtle distinctions between genuine and fabricated news content. These enhanced features are then processed by a sparsely structured Recurrent Neural Network (SRNN), which selectively focuses on the most salient information while modeling the sequential dependencies and flow within an article. This dual mechanism, semantic depth from WFT-BERT and temporal coherence from the SRNN, significantly improves the system's capacity to preserve critical patterns indicative of falsity. The robustness and generalizability of the proposed technique are rigorously validated through comprehensive evaluation across four diverse benchmark datasets: BuzzFeed, PolitiFact, Fakeddit, and Weibo.

Literature Review

This section presents an account of the existing models proposed for fake news detection, along with their advantages and limitations.

Kaliyar *et al*. (2021) implemented a DeepFakE model that employed tensor decomposition for fake news detection. Data on users' news engagement were collected and combined with user community data to construct a 3-mode tensor comprising context, content, and user community. To acquire latent representation of a news article, coupled matrix-tensor factorization was applied. DeepFakE, along with XGBoost, was then used for classification. This approach demonstrated improved performance by effectively integrating contextual and content-based features. The DeepFakE model was tested on the BuzzFeed dataset, with the decomposed factors were used as input features for classifying news. However, DeepFakE struggled to capture evolving patterns and deal with the dynamic nature of fake news due to its static structure.

Kaliyar *et al*. (2021) implemented DeepFakE by employing tensor decomposition. The data on users' news engagement were gathered and integrated with data from a user community to establish a 3-mode tensor, consisting of context, content, and user community. To acquire a news article's latent representation, coupled matrix-tensor factorization was performed. Then, DeepFakE and XGBoost were utilized for classification. This approach provided better performance by integrating context and content techniques. The DeepFakE method was effectively tested on the BuzzFeed dataset. The factors acquired after decomposition were used as features for classifying news. However, DeepFakE faced difficulties in capturing evolving patterns and the dynamics of fake news due to its static nature.

Che *et al*. (2024) presented a Sparse and Graph-Regularized CANDECOMP/PARAFAC (SGCP) learning approach for tensor decomposition in fake news detection. The news factor matrix was established through CP tensor decomposition, reflecting intricate associations between users and news. This approach used two datasets, BuzzFeed and PolitiFact, to evaluate the proposed method. The approach preserved the sparsity of the news factor matrix and maintained the structure of the manifold from the original space. Nevertheless, the SGCP learning approach struggled with scalability on large-scale social media data due to high computational complexity.

Palani *et al*. (2022) developed CB-Fake for the detection of fake news. BERT was employed to extract textual features that preserved the semantic relationships among words. CapsNet was utilized to capture visual features from images. These features were integrated to acquire a richer representation of the data for determining whether the news was real or fake. The CB-Fake approach was efficient and scalable for detecting fake news. However, CB-Fake relied on surface-level features and predefined patterns, which missed nuanced language use and failed to identify sophisticated fake news.

Shishah (2021) introduced BERT with joint learning, integrating Named Entity Recognition (NER) and Relational Features Classification (RFC) to detect fake news. SPR-encoder modified k-layer dynamic attention range in BERT, was used to establish the context vector by employing prior knowledge in the provided pre-trained technique. This BERT joint approach generated meaningful weights for features, thereby providing better performance. However, BERT with joint learning struggled with ambiguous or context-dependent cues that were subtle or contradictory in fake news. This ambiguity made it challenging to accurately identify and classify entities, leading to reduced detection performance.

Kaliyar *et al*. (2021) implemented a deep learning approach based on Echo Chambers (EchoFakeD) using both context and content data for detection of fake news. Moreover, the model was designed with a distinct number of filters over dense layer with dropout. To categorize these data, a Deep Neural Network (DNN) was utilized with optimal hyperparameters. EchoFakeD method was validated on the BuzzFeed and PolitiFact datasets, achieving higher validation accuracy. Employing tensor decomposition in the implemented technique provided better performance. However, integrating both content and context data in the EchoFakeD technique struggled with generalization issues.

Wang *et al*. (2022a) developed a multi-EDU structure to enhance text representation for fake news detection, namely EDU4FD. The former was obtained by modeling the coherence among consecutive EDUs with TextCNN, which reflected semantic coherence. The rhetorical relations were extracted to construct the EDU dependency graph, and then a Relation Graph Attention Network (RGAT) was used to obtain a graph-based EDU representation. EDU4FD then concentrated on the primary EDU to establish a text representation for prediction. However, EDU4FD suffered from potential loss of contextual coherence, which led to misinterpretation.

Liu *et al*. (2023) suggested an image caption-based technique to improve semantic data from images for fake news detection. The image description data were integrated into text to bridge the semantic gap between images and text. Then, a transformer was employed to fuse multi-modal content. The object and global features from images were combined, which increased image utilization and improved the semantic interaction between text and images. However, the suggested approach depended on a predefined vocabulary, which struggled to accurately describe complex visual concepts.

Wang *et al*. (2022b) presented a Fine-grained Multimodal Fusion Network (FMFN) for fake news detection. A Deep Convolutional Neural Network (CNN) was used to extract various visual features from images. Fused feature was passed through a binary classifier for detection. Scaled dot-product attention not only considered the correlation among visual features but also captured the dependency among textual and visual features. However, FMFN suffered from overfitting due to the complex fusion process, which led to biased representations and decreased generalization performance on unseen data.

From overall analysis, it is noted that existing methods like DeepFakE, SGCP, CB-Fake, BERT with NER, EchoFakeD, multi-EDU, and FMFN demonstrate notable advancements in fake news detection, but still face critical limitations. These limitations include difficulties in adapting to evolving patterns and capturing the dynamics of fake news, struggles with scalability on large-scale social media, challenges with ambiguous or context-dependent cues, and generalization issues. Additionally, the rapid and extensive spread of fake news and rumors worldwide has led to the production and propagation of inaccurate news articles. These challenges highlight the need for a more robust, effective, and context-sensitive method. In order to address this issue, WFT-BERT-SRNN is proposed for accurate news detection by integrating the contextual strength of BERT with the exceptional efficiency of SRNN. BERT captures deep semantic meaning, ensuring contextual coherence and minimizing misinterpretation. WFT-BERT enhances the model’s ability to understand nuanced language by assigning different importance weights to phrases and words, increasing sensitivity to subtle and context-dependent cues. This fine-tuning enables the model to better adapt to evolving patterns in fake news. Meanwhile, SRNN effectively captures long-range dependencies and temporal dynamics in news propagation, addresses scalability issues by concentrating on the most relevant features, and minimizes computational complexity. Together, WFT-BERT and SRNN provide a robust framework that generalizes well and enhances accuracy in detecting fake news by effectively modeling both temporal evolution and linguistic complexity.

Methods

The WFT-BERT-SRNN is proposed for fake news detection in this research. Initially, data is acquired from the BuzzFeed, PolitiFact, Fakeddit, and Weibo benchmark datasets to evaluate the performance of the WFT-BERT-SRNN technique. Data pre-processing is performed using stop word removal, tokenization, and stemming techniques to eliminate unwanted phrases or words. WFT-BERT is utilized to extract features, while SRNN is used to detect fake news as real or fake. Fig. 1 illustrates a block diagram of proposed technique for fake news detection.

Datasets

Fake news detection is evaluated on four benchmark datasets, Buzzfeed, PolitiFact, Fakeddit and Weibo. A detailed description of these datasets is given below.

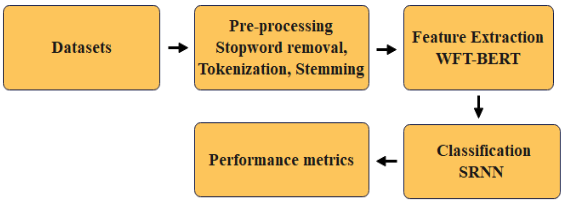

Fig. 1: Block diagram illustrating the workflow of proposed fake news detection representing stages like pre-processing, feature extraction, and classification

The Buzzfeed dataset contains two types of news: real and fake. The data is gathered from articles related to fake news about the 2016 US Presidential Elections. It contains 1700 articles gathered from Facebook. This dataset includes 182 news articles, of which 91 are fake news articles and the remaining are real, with approximately 15,257 users. The dataset includes terms such as 'nation', 'country', 'party', 'political', 'democrat', 'bill', among others.

PolitiFact contains 14,055 articles provided by 3,634 authors, with an average of 3.86 articles per author. These articles cover 52 subjects, with each article covering more than one subject. The dataset includes 240 news articles, of which 120 are fake news articles and the rest are real, reaching 23,865 users. Moreover, each article has a credibility score that takes one of two values: false or true.

The Fakeddit dataset is comprised of posts from Reddit, a popular social media platform. It contains over 1 million samples with multi-grained labels, covering metadata, text, comments, and images. The dataset provides labeling information for 2, 3, and 6 classes, offering granularity for classification.

The Weibo dataset originates from China's popular social media platform, Weibo. Each news item contains a corresponding image, label, and text data. Overall, the data are split into 80% training and 20% testing sets. The obtained data are then fed into the pre-processing stage.

Pre-Processing

After data acquisition, pre-processing is performed by employing stop word removal, tokenization, and stemming, as discussed below (Qorib *et al*., 2023).

Tokenization

Tokenization is the process of dividing the original text into smaller segments called tokens. Punctuation in the text is eliminated using this approach. Number filters are utilized to remove numeric terms from sentences. Case converters transform the textual data into either upper or lower case. Finally, words with fewer characters are eliminated using N-char filters.

Stop Word Removal

Stop words are not crucial to the meaning of a sentence, but they are frequently used in the connection and completion of expressions. They are common in almost every sentence and do not carry significant information. In English, there are about 500 stop words, including conjunctions, prepositions, and pronouns, which are regarded as typical stop words. Examples include "a," "when," "on," "what," "am," "an," and "under." Removing stop words helps save both processing time and storage space.

Stemming

The primary aim of the stemming process is to obtain the root form of words that carry the same meaning but differ in their inflectional forms. Various grammatical forms such as adjectives, adverbs, verbs, and nouns are reduced to their root form during this process. For instance, the words "consulting" and "consultants" are stemmed from the word "consult." Reducing words to a standard base form is considered an efficient technique. Hence, redundant and unnecessary terms such as extra text, numbers, and stop words are filtered during the pre-processing stage before being passed on for feature extraction.

Feature Extraction (FE)

The pre-processed data is fed as input into the WFT-BERT for extracting features used in the detection of fake news (Viji and Revathy, 2022). WFT refers to a refined version of traditional BERT fine-tuning, where each token is assigned a different level of importance based on its contribution to the classification task. Unlike conventional fine-tuning, which treats all tokens equally during training, WFT emphasizes semantically significant or deceptive tokens. This ensures the model pays more attention to phrases or words that are more likely to indicate fake news. Through a process of token replacement and insertion, the model determines which words most influence the classification and adjusts attention weights accordingly. Text sequences are indicated as $A = \{a_1, \dots, a_L\}$, where $a_1 (1 \leq 1 \leq L)$ represents the tokens in a sentence, and L indicates the length of the text sequences. Using a bidirectional pre-trained approach, the sequence A is encoded into a fixed-length sentence vector h, which serves as the input source-element. The sentence vector s_1 is indicated in Eq. 1:

$$s_1 = BERTsent(a_1) \tag{1}$$

Where, $BERTsent(.)$ represents encoding of sentences into sentence vectors. Hidden vector representation u_1 of the transferred sentence vector s_1 is obtained by employing a Multi-Layer Perception (MLP). The mathematical formula for u_1 is expressed in Eq. 2:

$$u_1 = tanh(W_1 s_1 + b_1) \quad (2)$$

Where, W_1 and b_1 indicate weight and bias parameters. Traditional text representation approaches often eliminate interaction information among text sentences, resulting in the loss of partial semantics. In this context, all source elements are considered as contextual information to obtain a text representation that retains richer semantics. For example, let c be one of the source elements h_k, and all source elements $(s_1, s_2, \dots, s_L)$ are captured. The semantic weight a_1 is assigned using a_k, where the source element is expressed as α_{k1}, as represented in Eq. 3, and i_k is expressed in Eq. 4:

$$\alpha_{k1} = \frac{exp(ul^k u_k)}{\sum_{1=1}^{L} exp(u_1^T u_k)} \quad (3)$$

$$i_k = \sum_{1-1}^{L} \alpha_{k1} S_1 \quad (4)$$

Each element of a single source interacts with all other source elements, capturing the interactions between all source elements and the individual source element, as indicated in Eq. 5:

$$I = (i_1, i_2, \dots, i_L) \quad (5)$$

The interaction contributes to the final representation of unequal text and an attention layer is included to enable data interaction, which follows a process similar to classification. Here, s indicates the compatibility score with respect to the weight I, and I represents the interaction representation. In joint word embedding process, compatibility score of entire text is generated. Therefore, final text T is expressed in Eq. 6:

$$T = sI \quad (6)$$

Each sentence is indicated as $a_1 = \{wd_1, wd_2, \dots, wd_n\}$, such that wd_1 indicates each word in a sentence. BERT approach encodes all sentences $a_1 = \{wd_1, wd_2, \dots, wd_n\}$ to their base form of respective word embedding $\{E_1, \dots, E_n\}$, which is denoted in Eq. 7:

$$V_1 = BERTtoken(a_1) \quad (7)$$

Where, $BERTtoken(a_1)$ indicates the encoding of word into their corresponding word vectors. Word embedding representation of the whole text sentence $A = \{a_1, \dots, a_L\}$ is denoted as $E = \{E_1, E_2 \dots, E_1\} = \{\{e_1, \dots, e_n\}, \{e_1, \dots, e_n\}, \dots, \{e_1, \dots, e_n\}\}$, where n represents the overall count of words. In addition, b refers to the associating sequence of text labels. The text label sequence A is encoded into its label embedding form $F = \{f_1, f_2 \dots, f_k\}$ evaluated by BERT, where, K denotes the number of classes. The embedding representation $F = \{f_1, f_2 \dots, f_k\}$ is expressed in Eq. 8:

$$f_k = BERTtoken(b) \quad (8)$$

The labels and words are embedded into a unified joint space. Cosine similarity is used to compute the compatibility G between label–word pairs, as denoted in Eq. 9:

$$G = (F^T E) \emptyset \hat{G} \quad (9)$$

The division of element-wise aid in matrix or vector operations is indicated as $\emptyset$, and $\hat{G}$ denotes the normalized matrix of the size $K \times L$. All elements of the normalized matrix are expressed as $\hat{g}||ck|| \; ||el||$, where $||.||$ indicates the $L2$ norms. Here, e_1 and f_k represent the 1^{th} word and k^{th} label embedding, respectively. By employing non-linear function, spatial relative data among consecutive words are computed by capturing the compatibility of label word pairs. In Eq. 10, e_q indicates the stigmatization of high-level compatibility between the labels and the q^{th} phrase:

$$e_q = ReLU(G_{q-i:q+i} WRD_2 + b_2) \quad (10)$$

Where, $G_{q-i:q+i}$ represents the label-to-token compatibility, WRD_2 denotes the weight, and b_2 indicates bias. Maxpooling operation acquires greatest compatibility value in the q^{th} phrase according to the entire labels, as expressed in Eq. 11:

$$m_q = max(e_q) \quad (11)$$

The whole sequence of the text's compatibility score is expressed in Eq. 12. Where, s_q indicates the q^{th} element of Softmax, as expressed in Eq. 13:

$$s = Softmax\,(m) \quad (12)$$

$$s_q = \frac{exp(m_q)}{\sum_{q^1}^{L} exp(m_q)} \quad (13)$$

Where, L and m represent the length and vector. The entire text sequence's compatibility score s is computed by employing words learning embedding and label embedding, s is utilized to capture large interactive-data and weigh the representation of finalized interactive text $I = (i_1, i_2, \dots, i_L)$. Moreover, the labels learn from a large volume of textual data, and the classifier uses these weighted labels effectively for classification. Additionally, s is utilized to weigh the final vector of the label F_k, which is expressed in Eqs. 14 and 15:

$$T = \sum q s_Q i_q \tag{14}$$

$$\acute{F} = \sum q s_Q F_k \tag{15}$$

Where, T and $\acute{F}$ indicate the final text and label representations. Also, the similarity scores of various statements is extracted in the analysis stage, and trained classification is indicated as $C: S \lozenge Y$. A setting of the soft-label is considered such that the attacker queries classifier to acquire the output probabilities from the generated input. Model parameters and training data does not provide for access. For instance, the weight example is represented as S_weight, which needs to be generated for the given input-pair condition $C(S_{weight}) \neq y$, where this condition must be satisfied. The S_weight must be grammatically accurate with S semantics. There are two types of token-level perturbations: new token insertion and token replacement, both established for generating a weighted example. This process is carried out in two steps: i) Replacement of token of condition $a \in S$ with another token, and ii) Insertion of a new token a with S.

Fewer input tokens contribute more to the final detection via C, compared to others. The replacement of tokens or the insertion of a new token has a stronger impact on updating the classifier's detection ability. The I_i token's importance is computed for each 'a' via removing 'a' from S, also by reducing the probability computation in the detection of the correct label (y). The pre-trained BERT technique is employed for the detection of similar tokens. These similar tokens are well-suited for grammar and text context. During token replacement, if multiple-tokens occurrences exist, they cause misclassification C for S. The token that enables S_weight more similar to the original S, based on the similarity score, is then selected. If no misclassification occurs, another token that reduces the detection probability is chosen. Token perturbation is employed iteratively until either $C(S\_weight) \neq S$ or all tokens in S are perturbed. BERT has a limited ability to model long-range dependencies due to its fixed context size. Therefore, BERT is used for the extraction process, not for classification. After feature extraction, classification is performed using SRNN, which has the potential to capture long-range dependencies in semantic data.

Classification

After extracting the features, the SRNN is employed to effectively detect and classify fake news. The extracted features from WFT-BERT are input into a Sparse RNN by converting the dense BERT embeddings into sparse representations. SRNN is preferred over conventional methods such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Swim Transformer because of its enhanced computational efficiency and minimized memory usage, making it well-suited for fake news articles. Unlike LSTM and GRU, which process each connection densely, SRNN selectively activates only significant neurons, resulting in faster training and inference. Moreover, it reduces overfitting by introducing sparsity in the connections. Compared to the Swim Transformer, SRNN manages temporal dependencies more effectively with fewer parameters, making it ideal for sequential text classification. This allows the SRNN to process sequential data within the extracted features, and detect and classify based on the learned temporal dependencies. Moreover, sparse refers to sparsity of neuron-to-neuron connections. Unlike traditional RNN or LSTM which are densely connected and computationally intensive, SRNN utilizes sparse topology which means a subset of possible neural connections are active. This is achieved via Erdos-Renyi random initialization, dynamic pruning, and regrowing of connections during training. Such sparsity minimizes memory consumption, increases computational efficiency, and solves overfitting issues by concentrating on the most relevant patterns in sequential data. Hence, the proposed method establishes a robust and effective model capable of detecting fake news with high accuracy while maintaining low computational complexity.

RNN is a widely used neural network model in Natural Language Processing (NLP) because it retains prior computations to influence current outputs (Dasu *et al*., 2023). Unlike conventional neural networks that treat all inputs independently, RNNs pass information from one step to the next, making each output dependent on previous inputs, hence the term "recurrent." However, RNNs often face slow and complex training processes for classification tasks. To overcome these limitations, SRNN is introduced, which enhances generalization by learning relevant patterns and connections while improving efficiency and effectiveness during both training and inference. This approach employs a sparse training method, establishing SRNN as a new class of RNN models.

Sparse Topology Initialization

In sparse topology initialization, the network y is expressed in Eq. 16:

$$y = f(x; \theta) \tag{16}$$

Where, $\theta \in R$ indicates the network's dense parameter. Instead of beginning with a dense parameter, this technique makes the network begin with θ_s. Here, to enforce a sparse structure, the masks are employed due to its limited support for sparse connections. The network is initialized in the Eq. 17:

$$\theta_s = \theta * M \tag{17}$$

Where, M represents a binary mask in which nonzero components are initialized by Erdos-Renyi or random distribution. Erdos-Renyi is established where M_{ij}^{k} between the neuron h_{j}^{k-1} and h_{i}^{k} exist with probability, as indicated in Eq. 18:

$$PM_{ij}^{k} = \frac{\epsilon(n^{k}+n^{k-1})}{n^{k}n^{k-1}} \tag{18}$$

Where, n^{k}, n^{k-1} indicates the number of neurons of h^{k} and h^{k-1}, and ϵ represents the parameter evaluated at the level of sparsity s. Initialized by the topology of Erdos-Renyi, layers with the greatest weights have larger sparsity than the smaller ones. Sparse initialization is another technique that initializes each layer with identical sparsity with overall sparsity s.

Pruning Strategy

In Sparse Evolutionary Tracking (SET), unlike magnitude-based pruning, a different approach is employed that eliminates the ς function of the smallest positive and largest negative weights of each layer after training at each epoch. Another pruning variant selects weights with the smallest absolute values. For each θ_{s}^{i}, its significance is defined as its absolute value, which is expressed in Eq. 19:

$$S\theta_{s}^{i} = |\theta_{s}^{i}| \tag{19}$$

The p^{th} percentile of $S(\theta_{S})$ is determined by the pruning rate p in ascending order γ. Then, the new mask is expressed in Eq. 20. Additionally, during training, the pruning rate p is decayed iteratively to 0, so that the topology of sparse converges to the optimal one:

$$M = S(\theta_{S}) > \gamma \tag{20}$$

Regrowing Strategy

The new weights are randomly regrown by employing data of non-zero parameters to maintain a pure sparse structure, both for backward and forward procedures. This is the primary difference among ST-RNN with gradient-based methods of sparse training like Sparse Networks from Scratch (SNFS) and Rigged Lottery (RigL). Gradient-based regrowing depends greatly on all gradient's parameters, and still needs a dense forward pass, with at least once per ΔT iterations, while SRNN maintains clearly backward sparse pass and needs lesser Floating-Point Operations (FLOPs). Random regrow is mathematically expressed in Eq. 21:

$$M = y + R \tag{21}$$

Where, R represents the binary tensor, and non-zero components are distributed randomly. The overall number of newly activated associations equals the number of eliminated associations to maintain a consistent level of sparsity. Furthermore, each layer's sparsity level is kept fixed. The FLOPs required to train the model are proportional to those of its dense counterpart. As the sequential data are generated in SRNN with varying time steps, the model effectively captures long-range dependencies. It also processes sequences of sparse input with appropriate context to detect fake news patterns. Through classification, the model differentiates genuine from misleading data based on the learned sequential patterns, thereby enhancing accuracy and effectively detecting fake news. The hyperparameters of SRNN include a learning rate of 0.001, which provides a stable balance between convergence speed and performance. The Adam optimizer is chosen for its adaptive learning capabilities. The Rectified Linear Unit (ReLU) activation function and a dropout rate of 0.2 are applied to prevent overfitting and ensure non-linearity in feature learning, respectively. Fig.2. shows detailed architecture of WFT-BERT-SRNN indicating integration of WFT-BERT with SRNN for effective fake news detection.

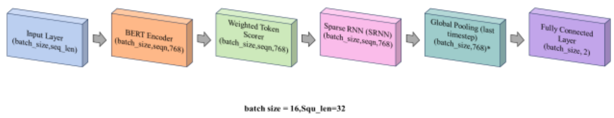

Fig. 2: Detailed architecture of WFT-BERT-SRNN method indicating integration of WFT-BERT with SRNN for effective fake news detection

Algorithm 1 shows the pseudocode of WFT-BERT-SRNN to improve clarity and reproducibility, while Table 1 presents the description of notations and symbols used throughout proposed methodology including variables.

Algorithm 1

Input: Datasets (Buzzfeed, PolitiFact, Fakeddit, and Weibo)
Output: Classification labels (Real, Fake)

Step 1: Pre-processing
For each article A in dataset D do
  Apply tokenization on A
  Remove stop words from A
  Perform stemming to minimize words to its root forms
End for

Step 2: Feature extraction by utilizing WFT-BERT
For each preprocessed article A do
  Encode sentence -level vectors using fine-tuned BERT
  Apply weighted label-to-token for capturing semantic importance
  Establish feature matrix F with semantic and contexual representation
End for

Step 3: Initialize SRNN
Establish sparse topology by applying Erdos-Renyi initialization
Generate binary mask M to enfprce sparsity on weight matrix W
Initialize sparse network weights

Step 4: Train SRNN for classification
For each training epoch do
Apply feature matrix F into SRNN
Calculate output prediction and classification loss using true labels
Establish backpropagation via sparse network
Prune weights with smallest absolute values
Regrow new random sparse connections to manage sparsity level
End For

Step 5: Fake news classification
For each test sample do
Input WFT-BERT features into trained SRNN
Predict label L ϵ $\{Real, Fake\}$
End For

Return: Predicted labels for test dataset

Table 1: Notation description and symbols employed throughout the proposed methodology section including variables

Symbol	Description
A	text sequences
$BERTsent(.)$	encoding sentences into sentence vectors
u_1	hidden vector representation
W_1 and b_1	weight and bias parameter
s	compatibility score
T	Final text
$BERTtoken(a_1)$	word embedding to their word vectors
n	overall count of words
K	class count
$\emptyset$	Matrix or vector operation
$\hat{G}$	normalized matrix
f_k	1^{st} word and k^{th} label embedding
e_q	stigmatization of high-level compatibility between entire labels and q^{th} phrase
$G_{q-i:q+i}$	label-to-token compatibility
WRD_2	weight
b_2	bias
s_q	q^{th} element of Softmax
L and m	length and vector
T and $\acute{F}$	final text and label representation
S_weight	weight example
y	network
$\theta \in R$	network's dense parameter
M	binary mask
n^k, n^{k-1}	number of neurons of h^k and h^{k-1}
ϵ	parameter evaluated the sparsity level s
p	pruning rate
γ	ascending order
R	binary tensor

Results

This section presents the performance results of WFT-BERT-SRNN, which is simulated in a Python 3.8 environment with system specifications of 16 GB RAM, an Intel Core i5 processor, and Windows 10 operating system. In detection and classification tasks, the accuracy score is utilized as a standard metric to evaluate model performance. However, accuracy alone is insufficient to fully analyze deep learning methods. A model with better performance should achieve higher scores across all performance metrics. Besides accuracy, this research employs recall, precision, and F1-score, as accuracy can be misleading when one class dominates. Precision measures the correctness of positive predictions, whereas recall captures the model's ability to identify all actual positives. The F1-score balances both precision and recall, providing a better measure of overall performance. These metrics assist in assessing how well the model performs. Relying solely on accuracy leads to poor performance on minority classes; hence, recall, F1-score, and precision are also calculated. These four metrics are represented in Equations (22) to (25):

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \tag{22}$$

$$Precision = \frac{TP}{TP+FP} \tag{23}$$

$$Recall = \frac{TP}{TP+FN} \tag{24}$$

$$F1-Score = \frac{2\times TP}{2\times TP+FP+FN} \tag{25}$$

Where, FP is False Positive, TP is True Positive, FN is False Negative, and TN is True Negative.

Performance Analysis

This section presents the qualitative and quantitative evaluation of WFT-BERT-SRNN, as shown in Tables 2 to 5. Table 2 displays the outcomes of different feature extraction methods evaluated on Buzzfeed dataset representing performance in terms of diverse metrics. Existing techniques such as Bag of Words (BoW), Word2Vec, Term Frequency-Inverse Document Frequency (TF-IDF), and BERT are compared with WFT-BERT. Figure 3 provides a graphical representation of feature extraction methods evaluated on the Buzzfeed dataset representing performance in terms of Precision, Accuracy, Recall, and F1-score. Compared to existing techniques, the WFT-BERT approach achieves a higher accuracy of 0.9847 due to its ability to capture deep contextual semantics with a fine-tuned mechanism tailored to specific features. Unlike BoW and TF-IDF, which eliminate word order and context, WFT-BERT leverages pretrained transformer layers that understand complex language patterns. While Word2Vec captures semantics, it lacks dynamic context representation.

DOI: 10.3844/jcssp.2025.2951.2964

Table 2: Performance analysis of different feature extraction methods utilizing Buzzfeed dataset to detect fake news

Performance Measures	BoW	Word2Vec	TD-IDF	BERT	WFT-BERT
Accuracy	0.8532	0.8614	0.8736	0.8771	0.9847
Recall	0.8425	0.8356	0.8547	0.8625	0.9704
Precision	0.8520	0.8453	0.8347	0.8598	0.9712
F1-score	0.8435	0.8374	0.8215	0.8603	0.9674

Table 3: Performance analysis of different classification methods based on Precision, Accuracy, Recall, and F1-score

Performance Measures	DNN	CNN	LSTM	RNN	RoBERTa	T5	XLNet	SRNN
Accuracy	0.9425	0.9547	0.9586	0.9635	0.8936	0.8746	0.8967	0.9847
Recall	0.9357	0.9468	0.9515	0.9596	0.9145	0.9268	0.9145	0.9704
Precision	0.9426	0.9536	0.9615	0.9654	0.9357	0.8896	0.9547	0.9712
F1-score	0.9357	0.9235	0.9567	0.9588	0.8769	0.9468	0.9353	0.9674
t-test	0.4392	0.5098	0.0009	0.0007	0.0746	0.1713	0.3537	0.0008

Table 4: Different feature extraction methods like BoW, Word2Vec, TF-IDF, BERT, and WFT-BERT using the PolitiFact datase which illustrates semantic and contextual representations

Performance Measures	BoW	Word2Vec	TD-IDF	BERT	WFT-BERT
Accuracy	0.8336	0.8426	0.8563	0.8625	0.9724
Recall	0.8416	0.8320	0.8436	0.8536	0.9612
Precision	0.8425	0.8303	0.8361	0.8584	0.9547
F1-score	0.8507	0.8436	0.8635	0.8502	0.9309

Table 5: Analysis of classification performance on PolitiFact dataset which represents enhanced performance by proposed method

Performance Measures	DNN	CNN	LSTM	RNN	RoBERTa	T5	XLNet	SRNN
Accuracy	0.9412	0.9456	0.9548	0.9625	0.8548	0.9147	0.8864	0.9724
Recall	0.9354	0.9375	0.9456	0.9520	0.8963	0.8715	0.9158	0.9612
Precision	0.9432	0.9357	0.9435	0.9468	0.9354	0.9256	0.9356	0.9547
F1-score	0.9135	0.9257	0.9215	0.9265	0.8634	0.9368	0.8975	0.9309
t-test	0.1042	0.2350	0.2790	0.5234	0.0264	0.3201	0.1748	0.0005

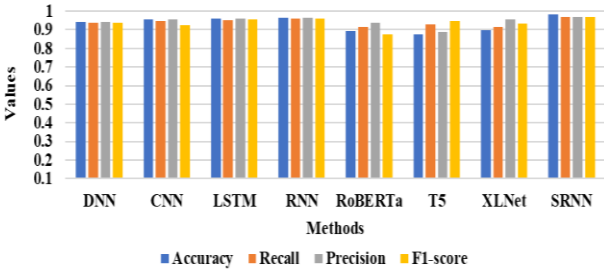

Fig. 3: Graphical representation of feature extraction methods evaluated on Buzzfeed dataset representing performance in terms of Precision, Accuracy, Recall, and F1-score

WFT-BERT establishes weighting mechanisms that emphasize informative tokens, thereby enhancing extraction performance.

Table 3 shows the different classification performance based on Precision, Accuracy, Recall, and F1-score using the Buzzfeed dataset. Existing techniques such as DNN, CNN, LSTM, RNN, Robust BERT (RoBERTa), Text-to-Text Transfer Transformer (T5), and XLNet are compared with the SRNN technique. Compared to these methods, SRNN achieves a high accuracy of 0.9847 and a t-test value of 0.0008 due to its ability to learn compact and effective representations through sparsely connected neurons, which minimize overfitting and improve generalization. SRNN selectively activates significant connections, enabling it to focus on important patterns in text data. When integrated with WFT-BERT, it captures rich contextual embeddings while maintaining computational efficiency. This ensures the model effectively captures both structural relevance and semantic depth. As a result, SRNN achieves high accuracy in fake news detection. Figure 4 provides a graphical representation of different classification performance using the Buzzfeed dataset for fake news. When compared to existing techniques, SRNN achieves an accuracy of 0.9847.

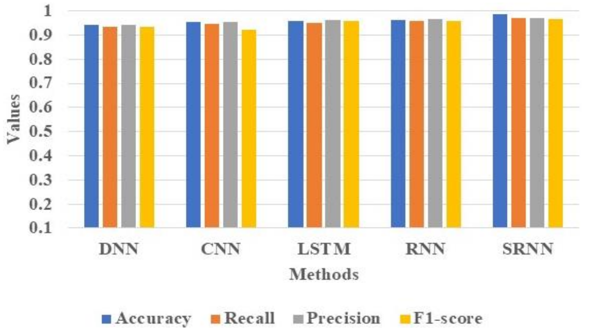

Fig. 4: Graphical representation of classification performance utilizing Buzzfeed dataset which represents superior performance of SRNN

Table 4 shows different feature extraction methods like BoW, Word2Vec, TF-IDF, BERT, and WFT-BERT using the PolitiFact dataset which illustrates semantic and contextual representations. Existing techniques such as BoW, Word2Vec, TF-IDF, and BERT are compared with WFT-BERT. Figure 5 provides a graphical representation of the different feature extraction methods like BoW, Word2Vec, TD-IDF, BERT, and WFT-BERT using PolitiFact. While BoW and TF-IDF rely on static features and eliminate word order and semantics, Word2Vec provides limited context and lacks fine-tuning for specific tasks. Moreover, WFT-BERT integrates contextual embeddings with a weighting mechanism, helping to focus on the most relevant features. This results in a better accuracy of 0.9724 compared to the existing techniques.

Table 5 shows the analysis of different classifier performance on PolitiFact dataset for fake news. Existing techniques such as DNN, CNN, LSTM, and RNN are compared with the SRNN technique. Figure 6 provides a graphical analysis of classification without Feature Extraction (FE) using PolitiFact with accuracy, f1-score, precision, and recall across different methods. Compared to these existing techniques, SRNN achieves superior accuracy of 0.9724 and a t-test value of 0.0005 by effectively capturing long-range dependencies while minimizing redundant information through sparsity. Its recurrent structure helps the model capture sequential patterns in text, which is crucial for understanding linguistic cues in fake news. Moreover, the sparsity mechanism improves generalization by preventing overfitting.

Table 6 presents a performance analysis of computational complexity per epoch in terms of time complexity, inference time, and memory consumption which demonstrates efficiency of SRNN in processing textural data. Compared to existing methods such as DNN, CNN, LSTM, RNN, RoBERTa, T5, and XLNet, SRNN achieves lower computational time complexity of 187 s and 192 s due to its ability to process only the most relevant connections, significantly minimizing the number of computations compared to the existing methods. By removing redundant weights, it reduces memory usage and inference time. This sparse connectivity enables rapid training and testing without significantly compromising accuracy. Moreover, the sparse model is easier to parallelize, enhancing efficiency compared to existing methods.

Comparative Analysis

Tables 7, 8, and 9 display a comparative analysis of existing techniques with proposed method on the Buzzfeed and PolitiFact datasets. The existing techniques by Kaliyar *et al.* (2021); Wang *et al.* (2022a) are compared with WFT-BERT-SRNN using the Buzzfeed dataset. The approaches by Kaliyar *et al.* (2021); Shishah (2021); Wang *et al.* (2022b) are used for comparison with WFT-BERT-SRNN on the PolitiFact dataset. The techniques by Liu *et al.* (2023); Wang *et al.* (2022) are compared with the proposed method using the Fakeddit and Weibo datasets. Compared to these existing techniques, the proposed WFT-BERT-SRNN achieves better accuracy, with values of 0.9847, 0.9724, 0.9624, and 0.9725 on the Buzzfeed, PolitiFact, Fakeddit, and Weibo datasets, respectively.

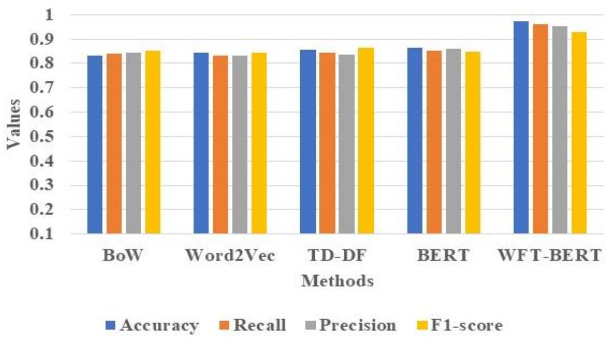

Fig. 5: Graphical representation of different feature extraction methods like BoW, Word2Vec, TD-IDF, BERT, and WFT-BERT on PolitiFact dataset

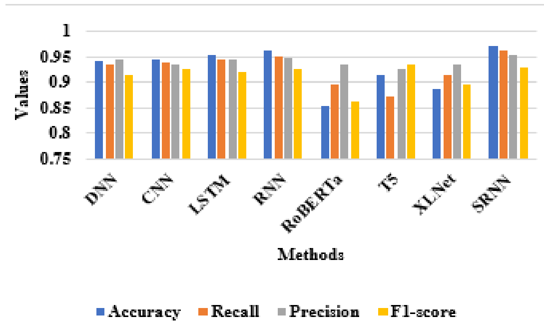

Fig. 6: Graphical analysis of classifier performances on PolitiFact dataset which shows accuracy, f1-score, precision, and recall across different methods

Table 6: Performance analysis of computational complexity per epoch in terms of time complexity, memory consumption, and inference time

Performance Measures	Datasets	DNN	CNN	LSTM	RNN	RoBERTa	T5	XLNet	SRNN
Time complexity (s)	Buzzfeed	269	215	193	236	248	263	198	187
Memory consumption (MB)		296	315	268	258	240	239	302	224
Inference time (ms)		425	410	396	378	465	403	380	358
Time complexity (s)	Politifact	265	278	219	239	245	198	236	192
Memory consumption (MB)		298	364	302	278	293	271	278	264
Inference time (ms)		398	365	378	375	389	364	378	349

Table 7: Comparative Analysis of existing techniques with proposed method utilizing Buzzfeed datasets

Performance Measures	DeepFake (Kaliyar *et al*., 2021)	EDU4FD (Wang *et al*. 2022)	Proposed WFT-BERT-SRNN
Accuracy	0.8649	0.7488	0.9847
Recall	0.8696	0.7486	0.9704
Precision	0.8333	0.7519	0.9712
F1-score	0.8511	0.7475	0.9674

Table 8: Comparative Analysis of existing techniques with proposed method utilizing PolitiFact datasets

Performance Measures	DeepFake (Kaliyar *et al*., 2021)	BERT-joint framework (Shishah, 2021)	EDU4FD (Wang *et al*., 2022)	Proposed WFT-BERT-SRNN
Accuracy	0.8864	0.84	0.7162	0.9724
Recall	0.8460	N/A	0.7111	0.9612
Precision	0.8210	N/A	0.7155	0.9547
F1-score	0.8404	0.87	0.7110	0.9309

Table 9: Comparative Analysis of existing techniques with proposed method using Fakeddit and Weibo datasets

Performance measures	Fakeddit dataset		Weibo dataset		
	Image caption-based technique (Liu *et al*. (2023)	Proposed WFT-BERT-SRNN	Image caption-based technique (Liu *et al*. 2023)	FMFN (Wang *et al*. 2022)	Proposed WFT-BERT-SRNN
Accuracy	0.9251	0.9624	0.8886	0.885	0.9725
Recall	0.9374	0.9515	0.9201	N/A	0.9615
Precision	0.9383	0.9520	0.8692	N/A	0.9536
F1-score	0.9379	0.9588	0.8939	N/A	0.9621

Discussion

The upsides of the proposed WFT-BERT-SRNN and the limitations of existing approaches are discussed. The limitations of existing techniques are as follows: DeepFake faces difficulties in evolving patterns and capturing dynamics in fake news due to its static nature (Kaliyar *et al*., 2021). In SGCP+SVM, when a large number of samples are lost, the technique's detection performance decreased (Che *et al*., 2024). The BERT-joint framework struggled to differentiate among various information types, which affected the model's classification performance (Shishah, 2021). EDU4FD suffered from potential loss of contextual coherence, leading to misinterpretation (Wang *et al*., 2022). The image caption-based technique relied on a pre-defined vocabulary, which struggled to accurately describe complex visual concepts (Liu *et al*., 2023). Therefore, the proposed technique overcomes these limitations. The WFT-BERT's deep bidirectional learning enables it to achieve superior results compared to conventional architectures. SRNN increases generalization by learning appropriate patterns and connections, and gains efficiency and effectiveness during both inference and training. By applying these techniques, WFT-BERT-SRNN achieves a better accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 for the Buzzfeed, PolitiFact, Fakeddit, and Weibo datasets, respectively, compared to existing techniques like DeepFake, the BERT-joint framework, EDU4FD, the image caption-based technique, and FMFN. However, while WFT-BERT-SRNN is highly accurate, it still misclassifies some fake news as real and real news as fake due to subtle linguistic manipulation or domain-specific phrasing in certain articles. For instance, satire or parody content closely resembles real headlines in the Buzzfeed dataset, which confuses the classifier. In PolitiFact, highly opinionated content is linguistically similar to factual statements, leading to misclassification. Fakeddit contains community-generated content where sarcasm and humor blur authenticity, resulting in both false positives and false negatives.

Similarly, the Weibo dataset consists of short and ambiguous posts, lacking sufficient context for WFT-BERT-SRNN embeddings to distinguish fake from real news. While class imbalance is a known challenge, the proposed WFT-BERT-SRNN generalizes effectively across the four datasets: Buzzfeed, PolitiFact, Fakeddit, and Weibo, without requiring data augmentation. The proposed WFT-BERT-SRNN addresses class imbalance by leveraging weighted loss functions, which penalize misclassification of minority classes more heavily. Fine-tuning BERT helps the model capture patterns, especially in fake news. The SRNN enhances generalization by concentrating on the most appropriate features while minimizing overfitting. Together, these techniques enable the proposed model to maintain robustness without relying on data augmentation. As a result, it consistently achieves superior performance, with accuracies of 0.9847, 0.9724, 0.9624, and 0.9725 on the Buzzfeed, PolitiFact, Fakeddit, and Weibo datasets, respectively, demonstrating robust detection performance even without explicit data augmentation.

Generalization Ability

The proposed WFT-BERT-SRNN demonstrates stronger generalization abilities across four benchmark datasets: Buzzfeed, PolitiFact, Fakeddit, and Weibo. These datasets vary in linguistic style, content origin (social posts, news articles, and political commentary), and text length, offering a diverse evaluation. The use of weighted fine-tuning in BERT enables the model to adapt effectively to dataset-specific features, while SRNN sparsity focuses on salient temporal patterns, enhancing adaptability. Furthermore, the model maintains consistently high-performance metrics across the four datasets, showcasing its robustness. This indicates that the proposed WFT-BERT-SRNN is not only accurate but also capable of generalizing to various content types.

Conclusion

In this research, the WFT-BERT-SRNN is proposed for fake news detection. WFT-BERT is employed to extract features, enabling the prioritization of specific domains or tasks during pre-training, which enhances the model's ability to capture domain-specific information. SRNN is utilized to detect and classify fake news effectively, improving generalization by learning appropriate patterns and connections, alongside enhancing efficiency and effectiveness in both inference and training. By integrating the deep contextual feature extraction abilities of WFT-BERT and SRNN, the proposed model outperforms existing methods across four benchmark datasets. Compared to existing techniques such as DeepFake and the image caption-based technique, WFT-BERT-SRNN achieves superior accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 on the Buzzfeed, PolitiFact, Fakeddit, and Weibo datasets, respectively. Beyond these empirical results, the proposed WFT-BERT-SRNN offers practical implications for enhancing fake news detection systems where accuracy and computational efficiency are significant. Its sparse model makes it suitable for integration into social media platforms and news verification tasks. In the future, attention-based ensemble techniques will be explored to enhance feature importance and context understanding. Moreover, the Secretary Bird Optimization Algorithm (SBOA) will be used to select the most appropriate features by optimizing model parameters. These techniques are expected to enhance detection accuracy and convergence in complex problem spaces like fake news detection.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

This research received no external funding.

Authors Contributions

Asha K: Conceptualization and formulation of the research problem**.** Design and development of the weighted fine-tuning approach for BERT-based Sparse RNN.

Mukta Pujar: Implementation of the proposed method, including training and finetuning the models.

Development and integration of data preprocessing pipelines for fake news datasets.

Akshatha AMS: Analysis and interpretation of experimental results.Design and execution of performance evaluation metrics.

Shilpa RN: Literature review and identification of research gaps in fake news detection techniques.

Writing the introduction and related work sections of the manuscript.

Shivaranjani S Shirabadagi: Review and refinement of the manuscript, ensuring clarity and coherence.

Managing correspondence with the journal and addressing reviewer comments.

Ethics

The author affirms that this study was conducted ethically, with all participants providing informed consent prior to data collection. No sensitive or personal data was disclosed, and all findings were reported with integrity and transparency. The author commits to addressing any ethical concerns that may arise following the publication of this manuscript.

Data Availability Statement

The datasets generated during and/or analysed during the current study are available in the Buzzfeed, Fakeddit, PolitiFact, and Weibo datasets [https: //www.kaggle.com/datasets/konradb/buzzfeed-news-2018-2023], [https: //github.com/entitize/Fakeddit], [https: //www.kaggle.com/datasets/subodh7300/politifact/code], and [https: //github.com/hlbao/weibo-public-opinion-datasets]

Conflicts of Interest

The authors declare no conflict of interest.

References

Albahar, M. (2021). A hybrid model for fake news detection: Leveraging news content and user comments in fake news. *IET Information Security*, *15*(2), 169–177. https://doi.org/10.1049/ise2.12021

Altheneyan, A., & Alhadlaq, A. (2023). Big Data ML-Based Fake News Detection Using Distributed Learning. *IEEE Access*, *11*, 29447–29463. https://doi.org/10.1109/access.2023.3260763

Aslam, N., Ullah Khan, I., Alotaibi, F. S., Aldaej, L. A., & Aldubaikil, A. K. (2021). Fake Detect: A Deep Learning Ensemble Model for Fake News Detection. *Complexity*, *2021*(1). https://doi.org/10.1155/2021/5557784

Che, H., Pan, B., Leung, M.-F., Cao, Y., & Yan, Z. (2024). Tensor Factorization With Sparse and Graph Regularization for Fake News Detection on Social Networks. *IEEE Transactions on Computational Social Systems*, *11*(4), 4888–4898. https://doi.org/10.1109/tcss.2023.3296479

Dasu, M. V., Reddy, P. V. N., & Reddy, S. C. M. (2023). Deep concatenated features with improved heuristic-based recurrent neural network for hyperspectral image classification. *Multimedia Tools and Applications*, *83*(17), 49875–49904. https://doi.org/10.1007/s11042-023-17351-0

Dixit, D. K., Bhagat, A., & Dangi, D. (2022). Automating fake news detection using PPCA and levy flight-based LSTM. *Soft Computing*, *26*(22), 12545–12557. https://doi.org/10.1007/s00500-022-07215-4

Elsaeed, E., Ouda, O., Elmogy, M. M., Atwan, A., & El-Daydamony, E. (2021). Detecting Fake News in Social Media Using Voting Classifier. *IEEE* Access, 9, 161909–161925. https://doi.org/10.1109/access.2021.3132022

Jain, M. K., Gopalani, D., & Meena, Y. K. (2024). ConFake: fake news identification using content based features. *Multimedia Tools and Applications*, *83*(3), 8729–8755. https://doi.org/10.1007/s11042-023-15792-1

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). DeepFakE: improving fake news detection using tensor decomposition-based deep neural network. *The Journal of Supercomputing*, *77*(2), 1015–1037. https://doi.org/10.1007/s11227-020-03294-y

Kaliyar, R. K., Goswami, A., & Narang, P. (2021). EchoFakeD: improving fake news detection in social media with an efficient deep neural network. *Neural Computing and Applications*, *33*(14), 8597–8613. https://doi.org/10.1007/s00521-020-05611-1

Kumar, S., Kumar, A., Mallik, A., & Singh, R. R. (2024). OptNet-Fake: Fake News Detection in Socio-Cyber Platforms Using Grasshopper Optimization and Deep Neural Network. *IEEE Transactions on Computational Social Systems*, *11*(4), 4965–4974. https://doi.org/10.1109/tcss.2023.3246479

Liao, Q., Chai, H., Han, H., Zhang, X., Wang, X., Xia, W., & Ding, Y. (2022). An Integrated Multi-Task Model for Fake News Detection. *IEEE Transactions on Knowledge and Data Engineering*, *34*(11), 5154–5165. https://doi.org/10.1109/tkde.2021.3054993

Liu, P., Qian, W., Xu, D., Ren, B., & Cao, J. (2023). Multi-Modal Fake News Detection via Bridging the Gap between Modals. *Entropy*, *25*(4), 614. https://doi.org/10.3390/e25040614

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*, *1*(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

Ozbay, F. A., & Alatas, B. (2021). Adaptive Salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media. *Multimedia Tools and Applications*, *80*(26–27), 34333–34357. https://doi.org/10.1007/s11042-021-11006-8

Palani, B., Elango, S., & Viswanathan K, V. (2022). CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT. *Multimedia Tools and Applications*, *81*(4), 5587–5620. https://doi.org/10.1007/s11042-021-11782-3

Qorib, M., Oladunni, T., Denis, M., Ososanya, E., & Cotae, P. (2023). Covid-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset. *Expert Systems with Applications*, *212*, 118715. https://doi.org/10.1016/j.eswa.2022.118715

Rai, N., Kumar, D., Kaushik, N., Raj, C., & Ali, A. (2022). Fake News Classification using transformer based enhanced LSTM and BERT. *International Journal of Cognitive Computing in Engineering*, *3*, 98–105. https://doi.org/10.1016/j.ijcce.2022.03.003

Rezaei, S., Kahani, M., Behkamal, B., & Jalayer, A. (2022). Early multi-class ensemble-based fake news detection using content features. *Social Network Analysis and Mining*, *13*(1), 16. https://doi.org/10.1007/s13278-022-01019-y

Saleh, H., Alharbi, A., & Alsamhi, S. H. (2021). OPCNN-FAKE: Optimized Convolutional Neural Network for Fake News Detection. IEEE Access, 9, 129471–129489. https://doi.org/10.1109/access.2021.3112806

Seddari, N., Derhab, A., Belaoued, M., Halboob, W., Al-Muhtadi, J., & Bouras, A. (2022). A Hybrid Linguistic and Knowledge-Based Analysis Approach for Fake News Detection on Social Media. *IEEE Access*, *10*, 62097–62109. https://doi.org/10.1109/access.2022.3181184

Shishah, W. (2021). Fake News Detection Using BERT Model with Joint Learning. *Arabian Journal for Science and Engineering*, *46*(9), 9115–9127. https://doi.org/10.1007/s13369-021-05780-8

Viji, D., & Revathy, S. (2022). A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi – LSTM model for semantic text similarity identification. *Multimedia Tools and Applications*, *81*(5), 6131–6157. https://doi.org/10.1007/s11042-021-11771-6

Wang, Y., Wang, L., Yang, Y., & Zhang, Y. (2022b). Detecting fake news by enhanced text representation with multi-EDU-structure awareness. *Expert Systems with Applications*, *206*, 117781. https://doi.org/10.1016/j.eswa.2022.117781

Wang, J., Mao, H., & Li, H. (2022a). FMFN: Fine-Grained Multimodal Fusion Networks for Fake News Detection. *Applied Sciences*, *12*(3), 1093. https://doi.org/10.3390/app12031093

Wu, L., Rao, Y., Zhang, C., Zhao, Y., & Nazir, A. (2021). Category-controlled Encoder-Decoder for Fake News Detection. *IEEE Transactions on Knowledge and Data Engineering*, *35*(2), 1–1. https://doi.org/10.1109/tkde.2021.3103833