Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 1, January 2026, pp. 331~343
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i1.pp331-343

Weighted fine-tuned BERT-based sparse RNN for fake news detection

Asha Kathigi, Gautam Vinod Nair, Kruthika Kadurahalli Raghu, Kavya Pujar Prakash, Meghana Deepak Duttargi
Department of Artificial Intelligence and Machine Learning, GM Institute of Technology, Davanagere, India

Article Info

Article history:

Received May 16, 2024
Revised Oct 13, 2025
Accepted Dec 13, 2025

Keywords:

Deep learning
Fake news
Natural language processing
Sparse recurrent neural network
Weighted fine-tuned-bidirectional representation for transformers

ABSTRACT

Fake news refers to misinformation or false reports shared in the form of images, articles, or videos that are disguised as real news to try to manipulate people’s opinions. However, detection systems fail to capture diverse features of fake news due to variability in linguistic styles, contexts, and sources, which lead to inaccurate identification. For this purpose, a weighted fine-tuned-bidirectional encoder representation for transformer-based sparse recurrent neural network (WFT-BERT-SRNN) is proposed for fake news detection using deep learning (DL). Initially, data is acquired from Buzzfeed PolitiFact, Fakeddit, and Weibo datasets to evaluate WFT-BERT-SRNN. Pre-processing is established using stopword removal, tokenization, and stemming to eliminate unwanted phrases or words. Then, WFT-BERT is employed to extract features. Finally, SRNN is employed to detect and classify fake news as real or fake. Existing techniques like deep neural networks for Fake news detection (DeepFake), BERT with joint learning, and multi-EDU structure for Fake news detection (EDU4FD), Image caption-based technique, and fine-grained multimodal fusion network (FMFN) are compared with WFT-BERT-SRNN. The WFT-BERT-SRNN achieves a better accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 for Buzzfeed, Politifact, Fakeddit, and Weibo datasets compared to existing techniques like DeepFake, BERT-joint framework, EDU4FD, Image caption-based technique, and FMFN.

Corresponding Author:

Asha Kathigi
Department of Artificial Intelligence and Machine Learning, GM Institute of Technology
Davanagere, India
Email: info@gmit.ac.in

1. INTRODUCTION

Fake news is one of the primary severe dangers to democracy, journalism, and global commerce with appropriate collateral harm [1]. The platform of social networks generates a significant channel for consumers to build, access, and share diverse information. The deployment of social media networks is enhanced as various users receive and look for the latest update at applicable times. Moreover, social media provides an opportunity for spreading countless fake and misleading data to a user, and such data has destructive consequences for the community [2]. Fake news significantly diffuses deeper and faster than the truth, causing severe negative impacts on both society and individuals [3]. Facebook is one of the most commonly used media platforms for spreading fake news, compared to Twitter, Google, and webmail like Gmail and Yahoo [4]. The concerns increased by it have expanded only with the rising time of people spend on social media, and thus is the only primary source of news for them [5]. Additionally, certain aggregators of official news intentionally spread false news to obtain popularity, earn money, or achieve certain objectives of politics. It enables fake news to be extensive due to its lack of control over the Internet and simplicity [6]. During the 2016 US election, numerous fake news instances were proclaimed to have spread on social media which contained a new Air Marshal nomination in India during the presidential elections [7]. The effectiveness of evaluating and enduring knowledge with others enables enticing online social networks. Moreover, the scattering of instantaneous data at a greater pace with small effort enables for widespread diffusion of false data like fake news [8]. Disinformation and misinformation are the two kinds of fake information. Disinformation is false data declared to mislead the public which has political, economic, and social impacts [9]. Furthermore, it is challenging to terminate fake news when already shared numerous times, on a large scale [10]. Thus, fake news spreading has exerted a negative impact on the ability of social, personal life, and political patterns [11]. Hence, fake news detection has become one of the important issues of social media [12]. Hence, an efficient detection technique is established in this study to avoid these evil intents that assist people not to fall into them [13], [14]. Recent researches on the detection of fake news via DL has attained impressive success by employing various news features of social media like user features, text data, and user feedback [15]. Human's ability to identify false data without special assistance is only 54% according to studies. Hence, it is needed for computerized real and fake news detection [16].

However, detection systems fail to capture the diverse features of fake news due to the variability in linguistic styles, contexts, and sources which lead to inaccurate identification of fake news. Kaliyar *et al.* [17] implemented a DeepFakE by employing tensor decomposition. The user's news engagement was gathered and integrated with data of a user community to establish a 3-mode tensor like 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 a classification task. This implemented approach provided better performances by integrating context and content techniques. However, DeepFake faced difficulties in evolving patterns and capturing dynamics of fake news because of their static nature. Che *et al.* [18] presented an Sparse and Graph-Regularized CANDECOMP/PARAFAC (SGCP) learning of tensor decomposition for fake news detection. The news factor matrix was established by CP tensor decomposition that reflected intricate associations between the user and news. This approach employs two datasets: Buzzfeed and PolitiFact News, to evaluate the presented approach. This approach retains news factor matrix sparsity and preserves the structure of the manifold from the original space. However, when a huge number of samples were lost, the presented technique decreased the detection performance.

Palani *et al.* [19] developed a CB-Fake for the detection of fake news. The BERT was employed to extract the textual features that preserved the relationship of semantics among words. The CapsNet was utilized for capturing the visual features from images. These features were integrated to acquire the representation of richer data for determining whether the news was real or fake. The CBFake approach was efficient and scalable for detecting fake news. But, the CB-Fake generated visual and textual features from news articles but the data of the user's profile and behavioral characteristics were not evaluated. Shishah [20] introduced a BERT with joint learning, integrating named entity recognition (NER) and relational features classification (RFC) for fake news detection. The SPR-encoder which modified the k layer's dynamic attention range in BERT was utilized to establish the context vector by employing prior knowledge in the provided pre-trained technique. This BERT joint approach's uniqueness generates a meaningful weight to features, thereby providing a better performance. However, this approach faced difficulties in differentiating among various information types which affected the classification. Kaliyar *et al.* [21] implemented a Deep Learning approach based on Echo chambers (EchoFakeD) with both context and content data for the detection of fake news. The content of the news was fused with a tensor to obtain a latent representation of both social context and news content data. To categorize these data, a deep neural network (DNN) was utilized with optimal hyper-parameter. Employing tensor decomposition in an implemented technique provided better performance. However, integrating both content and context data in an EchoFakeD technique struggled with generalization issues.

Wang *et al.* [22] developed a multi-EDU structure to enhance the text representation for the detection of fake news namely EDU4FD. Initially, the rhetorical relations were extracted to construct the EDU dependency graph and then relation graph attention network (RGAT) was set to obtain a graph-based EDU representation. At last, these two representations were combined as the enhanced text representation by utilizing a gated recursive unit with global attention approach for the detection of fake news. However, EDU4FD suffers from variability in linguistic styles, contexts, and sources. Liu *et al.* [23] suggested an image caption-based technique to improve an semantic data from images for fake news detection. The image description data was integrated into text to bridge a semantic gap among images and text. Then, the transformer was employed to fuse multi-modal content. The object and global features from an images were combined which increase image utilization and improves an semantic interaction among text and image. However, the suggested approach depends on pre-defined vocabulary which struggles to accurately describe complex visual concepts. Wang *et al.* [24] presented a fine-grained multimodal fusion network (FMFN) to fuse textual and visual features for the detection of fake news. The deep convolutional neural network (CNN) was utilized to extract various visual features of an image. The scaled dot-product attention was employed to increase visual and textual features and fuse them. At last, the fused feature passed through the binary classifier for detection. Scaled dot-product not only considers the correlation among visual features but also captures the dependency among textual and visual features. However, FMFN suffers from overfitting issue due to complex fusion process which leads to biased representation and decrease generalization performance on unseen data. From the overall evaluation, the existing techniques has limitations like facing challenges with generalization, variability in linguistic styles, contexts, and sources which lead to inaccurate identification of fake news. To address this issue, the WFT-BERT-SRNN is proposed to detect fake news accurately.

The basic contributions of this research are:

- WFT-BERT is utilized to extract the features that enable prioritizing specific domains or tasks during pre-training, and increase the model's capability to capture domain-specific information.
- SRNN is employed for detecting and classifying fake news as real or fake, and it has the potential to capture long-range dependencies in the semantic data.
- Four benchmark datasets namely, BuzzFeed, PolitiFact, Fakeddit, and Weibo, are established to evaluate the WFT-BERT-SRNN technique.

The remaining section is structured as follows: Section 2 discusses the proposed methodology. Section 3 details a weighted fine-tuned-bidirectional encoder representation for transformer-based sparse recurrent neural network, while Section 4 displays the results and discussion, and Section 5 indicates the conclusion.

2. PROPOSED METHOD

The WFT-BERT-SRNN is proposed for fake news detection in this research. Initially, the data is acquired from Buzzfeed, PolitiFact, Fakeddit, and Weibo benchmark datasets to evaluate the WFT-BERT-SRNN technique. Data pre-processing is established using stopword removal, tokenization, and stemming to eliminate unwanted phrases or words. WFT-BERT is utilized for extracting the features, and SRNN is performed for detecting and classifying fake news as real or fake. Figure 1 represents the block diagram for the proposed technique.

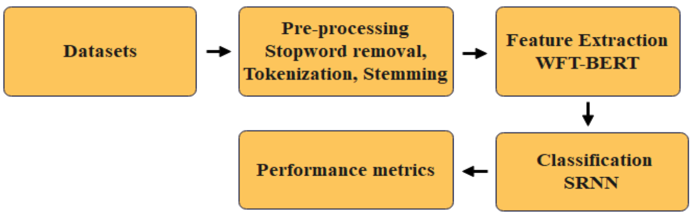

Figure 1. Block diagram for the proposed approach

2.1. Datasets

Fake news detection is evaluated by employing four benchmark datasets: Buzzfeed [25] and PolitiFact [26], Fakeddit [27], and Weibo [28]. Buzzfeed has two types of news sets that are real and fake news, and is gathered from the article of false news regarding a US-2016 presidential election. It contains 1700 articles of news gathered from Facebook. The chosen terms of this dataset are nation, country, party, political, democrat, bill, and so on. The PolitiFact dataset contains 14,055 articles that are provided by 3634 authors, who also have an average of 3.86 articles. These articles cover 52 subjects, and every article covers more than 1 subject. The obtained data are fed into the pre-processing stage.

The Fakeddit dataset is gathered from Reddit, which is a popular social media. It contains over 1 million samples and generates multi-grained labels which covers metadata, text, comments, and images. Weibo dataset arise from China's popular platform of social media. Each news item contains a corresponding image, label, and text.

2.2. Pre-processing

After data acquisition, the pre-processing is performed by employing stop word removal, tokenization, and stemming, which are discussed below. Tokenization removes punctuation from the text data, and stop word removal eliminates prepositions, pronouns, and conjunctions. Stemming removes grammatical words like adverbs, verbs, and so on.

- Tokenization: It is the procedure of original text division into smaller segments [29], which is described as tokens. Moreover, punctuation from text data is eliminated by using this approach. The number filters are utilized to eliminate the terms of numbers from a specific sentence. Case converters are employed for the textual data transformation to upper and lower cases. At last, the words including a lower number of characters are eliminated utilizing N-char filters.
- Stop word removal: It is inappropriate and not crucial but it is employed frequently in associating expressions and the completion of sentences. It is quite prevailed and usual in each sentence which does not carry any data. Moreover, in English, there are about 500 stop words that are conjunctions, prepositions, and pronouns which are regarded as few ‘stop words’. Instances of stop words are a, when, on, what, am, an, under, and so on. Hence, by removing the stop words, the processing time and space are saved.
- Stemming: The basic aim of a stemming [30] procedure is to obtain the basic form of words which carries the same meaning with different words. Numerous grammatical words like adjectives, adverbs, verbs, and nouns are converted into source form during this process. For instance, the words ‘consulting’ and ‘consultants’ stem from the word “consult”. Thus, the word reduction into a regular basic form is regarded as an efficient technique. Hence, the redundant and character terms namely, text, numbers, and stop words are filtered in the pre-processing stage which is passed for feature extraction.

3. WEIGHTED FINE-TUNED-BIDIRECTIONAL ENCODER REPRESENTATION FOR TRANSFORMERS-BASED SPARSE RECURRENT NEURAL NETWORK (WFT-BERT-SRNN)

3.1. Feature extraction

The pre-processed data is fed as input to the WFT-BERT [31] for extracting features for the detection of fake news. It has the ability to increase the representational capacity of model by emphasizing significant words or phrases which improves the detection accuracy. It employs pre-trained knowledge while being fine-tuned on specific data which ensures better contextual understanding. A text sequences are indicated as $A = \{a_1, \ldots, a_L\}$, where $a_1 (1 \leq 1 \leq L)$ points to sentences and L indicates the length of text sequences. By using bidirectional pre-trained approach, the text sequence is encoded as $A = \{a_1, \ldots, a_L\}$ to the vector of fixed-length sentence forms h, which is represented as input source-element. The sentence vector form s_1 is indicated in (1).

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

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

$$u_1 = \tanh(W_1 s_1 + b_1) \tag{2}$$

Where, W_1 and b_1 indicate the weight and bias parameter. The general utilized text-representation approach eliminates the interaction data between the text sentence which leads to the loss of partial semantics. Here, the whole source-elements are regarded as context-information to obtain the representation of text which contains more semantics. For example, c among one source element h_k and all source elements $(s_1, s_2, \ldots, s_L)$ are captured. The a_1 semantic weight is allocated by a_k, source element is expressed as α_{k1} which is represented in (3).

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

i_k is expressed in (4).

$$i_k = \sum_{1-1}^{L} \alpha_{k1} S_1 \tag{4}$$

Each element of single source has an interaction with all sources and acquires the interaction between all source-element and single-source elements as indicated in (5). The interaction devoted to a finalized representation of unequal text and attention layer is included to enable data-interaction which is similar to classification. The s indicates compatibility score to I weight and I represents the interaction representation. During joint word embedding process, the entire text’s compatibility score is produced. Therefore, the final text T is expressed in (6).

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

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

Every sentence is indicated as $a_1 = \{wd_1, wd_2, \dots, wd_n\}$, such that wd_1 indicates each word in a sentence. The pre-trained BERT approach encodes all sentences $a_1 = \{wd_1, wd_2, \dots, wd_n\}$ to their form of respective word embedding $\{E_1, \dots, E_n\}$, which is denoted in (7). Where, $BERTtoken(a_1)$ indicates the encoding of word to their word vectors. The representation of word-embedding of 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 points to associating sequence of text label A is encoded with their form of label embedding $F = \{f_1, f_2 \dots, f_k\}$ evaluated by BERT, where, K denotes the class count. $F = \{f_1, f_2 \dots, f_k\}$ is expressed in (8).

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

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

The labels and word are embedded to the space of one-joint. The simple technique to compute compatibility G between label word pairs is cosine similarity which is denoted in (9). The division of element-wise aid in matrix or vector operation which is indicated as $\emptyset$, $\hat{G}$ denotes the normalized matrix which consists of $K \times L$ size. All elements of the normalized matrix are expressed as $\hat{g}||ck||\ ||el||$, where $||.||$ indicates norms 2. e_1 and f_k represents the 1^{th} word and k^{th} label embedding. By employing non-linear function, the spatial relative data among consecutive words are computed during acquiring the compatibility of label word pairs. The (10) e_q indicates stigmatization of high-level compatibility between entire labels and q^{th} phrase.

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

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

Where, $G_{q-i:q+i}$ represents the label-to-token compatibility, WRD_2 denote weight, and b_2 indicate bias. The maxpooling operation acquires the value of greatest compatibility between q^{th} phrase according to the entire labels, as expressed in (11). The whole sequence of text's compatibility score is expressed in (12). The s_q indicates q^{th} element of Softmax which is expressed in (13).

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

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

$$s_q = \frac{\exp(m_q)}{\sum_{q1}^{L} \exp(m_q)} \quad (13)$$

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. The s is utilized to capture large interactive-data and to weigh the representation of finalized interactive text $I = (i_1, i_2, \dots, i_L)$. Moreover, the labels learn a large count of textual data, the classifier uses those weighted labels effectively for classification. Additionally, s is utilized for the weight finalized vector of label F_k, which is expressed in (14) and (15).

$$T = \sum q s_Q i_q \quad (14)$$

$$\acute{F} = \sum q s_Q F_k \quad (15)$$

Where, T and $\acute{F}$ indicates the final text and label representation. Also, the similarity scores of various statements is extracted in the analysis stage. The technique of trained classification indicates $C: S \lozenge Y$. The setting of soft-label is considered such that the attacker queries out the classifier to acquire the probabilities of output by the generated input. The model parameter and training data not provide to access. For instance, weight example is represented as S_weight, required to get generated for provided input-pair like condition $C(S_{weight}) \neq y$ need to be satisfied. The S_weight needs to be correct grammatically with semantic S. It has two kinds of token level perturbations of new token insertion and replacement of token, established for generating the weighted example S_weight. i) Replacement of token with condition $a \in S$ with another token. ii) The insertion of new token a with S.

Less input token contributes more for the final detection via C, when compared to others. The replacement of token or new token insertion has strong impacts in updating the classifier's detection. The I_i token importance is computed for each 'a' via removing 'a' from S, and also by reducing the probability computation in the detection of the correct label (y). The pre-trained BERT technique is employed for detection of similarity token. The similarity tokens suit well for grammatical text and text context. During the replacement of token, if there are multiple-tokens occurrences, they cause C for S misclassification, and then the token that enables S_weight more identical to the original S on the basis of similarity score is selected. If misclassification does not occur, another token which decreases the detection probability is selected. The perturbation of the token is employed iteratively till either $C(S\_weight) \neq S$ or whole S tokens are perturbed. BERT has a limited ability to model long-range dependencies because of the fixed context size. Hence, BERT is used as an extraction process, and not for classification. After feature extraction, the classification is performed using SRNN which has the potential to capture long-range dependencies in the semantic data.

3.2. Classification

After extracting the features, the SRNN is employed to detect and classify fake news detection effectively. The extracted features from WFT-BERT are input to a Sparse RNN by converting the embeddings of dense BERT into sparse representation. This enables the SRNN to generate sequential data involved in the extracted features, as well as detect and classify based on the learned temporal dependencies. The SRNN combines novel sparse connectivity with standard RNN architecture to effectively capture long-term dependencies. This approach preserves crucial information by maintaining key connections and eliminating redundant ones which ensures accurate detection. RNN [32] is a commonly utilized version of Neural Network employed in the natural language processing (NLP) because of its capability to remember the prior estimation. It contains every prior calculation to evaluate the present layer outcome. This contradicts the conventional neural network where all inputs are independent of each other; RNN's each input is transferred forward to the following layer, and the outcome of every layer is dependent on the prior layers. It is known as recurrent as it executes the same function for all components. However, the RNN has slow and complex training process for classification. To overcome this issue, the SRNN which increases the generalization by learning appropriate patterns and connection, and gains efficiency and effectiveness on both inference and training is employed. Here, a sparse training of RNN is introduced which is a new class of SRNN.

3.2.1. Sparse topology initialization

In sparse topology initialization, a network y is expressed in (16). Where, $\theta \in R$ indicates the network's dense parameter. Instead of begin with dense parameter, this technique makes the network begin with θ_s. Here, to enforce the structure of sparse, the masks are employed because of limited support for the connection of sparse. Basically, the network is initialized in the (17).

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

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

Where, M represents binary mask in which nonzero components. It is initialized by Erdos-Renyi or random distribution. Erdos-Renyi is established where the M_{ij}^k among neuron h_j^{k-1} and h_i^k exist with probability, as indicated in (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}, 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 as overall sparsity s.

3.2.2. Pruning strategy

In sparse evolutionary tracking (SET), different from magnitude-based pruning is employed that eliminates ς function of smallest positive and largest negative weights of every layer after training each epoch, another pruning of variant weight is selected with smallest absolute values. For each θ_s^i, its significance is defined as its absolute values which is expressed in (19). The p^{th} percentile of $S(\theta_S)$ is determined by some pruning rate p with ascending order γ. Then, the new mask is expressed in (20). Also, during training, the pruning rate p is decayed iteratively to 0, so that the topology of sparse converges to the optimal one.

$$S\theta_s^i = |\theta_s^i| \tag{19}$$

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

3.2.3. 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 procedure. It 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 at least once per ΔT iterations, while SRNN maintains clearly backward sparse pass and needs less floating-point operations (FLOPs). The random regrow is indicated in (21).

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

Where, R represents the binary tensor, and in that, the non-zero components are distributed randomly. The overall newly activated associations are same as the number of eliminated associations to maintain a level of same sparsity. Furthermore, each layer's sparsity level is kept fixed; the FLOPs required to train the model are proportional to their dense counterpart. As the sequential data are generated in SRNN with varying time steps, it enables the model to capture long-range dependencies effectively. It processes the sequence of sparse input, appropriate context and detecting fake news patterns. Furthermore, through classification, it differentiates among genuine and misleading data based on the learned sequential patterns, thereby enhancing the accuracy and detecting fake news effectively. Table 1 represents the Notation Description.

Table 1. Notation description

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

4. RESULTS AND DISCUSSION

Here, the WFT-BERT-SRNN is simulated by employing a Python 3.8 environment with 16GB RAM, Intel core i5 processor, and Windows 10 operating system. The accuracy, recall, f1-score, and precision are the parameters used to evaluate the proposed technique. The mathematical formula for these metrics are expressed in (22) to (25). Where, FP is false positive, TP is true positive, FN is false negative, and TN is true negative.

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \times 100 \tag{22}$$

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

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

$$F1 - Score = \frac{2 \times TP}{2 \times TP+FP+FN} \times 100 \tag{25}$$

4.1. Performance analysis

This section illustrates the qualitative and quantitative evaluation of WFT-BERT-SRNN which is represented in Tables 2 to 5. Table 2 shows different feature extraction methods using Buzzfeed. The existing techniques like bag of words (BoW), Word2Vec, term frequency-inverse document frequency (TD-IDF), and BERT are compared with WFT-BERT. Figure 2 represents the graphical representation of feature extraction analysis using Buzzfeed. The WFT-BERT approach achieves a better accuracy of 0.9847 when compared to these existing techniques because WFT-BERT captures contextual nuances and semantic relationships between words which provides richer representation. This enhanced understanding improves feature extraction and detection accuracy for fake news.

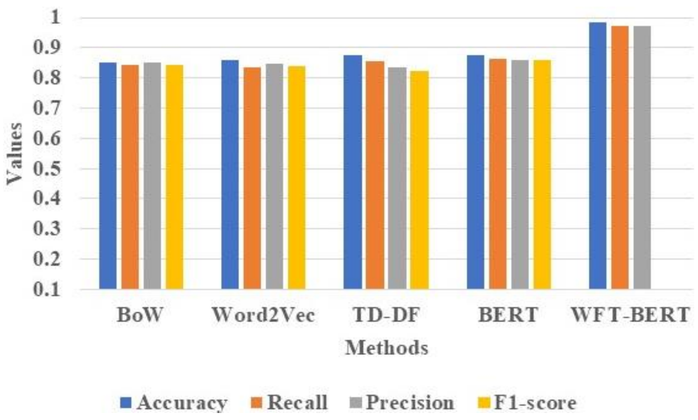

Figure 2. Graphical representation of feature extraction methods utilizing Buzzfeed

Table 2. Different feature extraction methods utilizing Buzzfeed dataset

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. Classification performance utilizing Buzzfeed dataset

Performance measures	DNN	CNN	LSTM	RNN	SRNN
Accuracy	0.9425	0.9547	0.9586	0.9635	0.9847
Recall	0.9357	0.9468	0.9515	0.9596	0.9704
Precision	0.9426	0.9536	0.9615	0.9654	0.9712
F1-score	0.9357	0.9235	0.9567	0.9588	0.9674

Table 3 shows the classification performances employing Buzzfeed. The existing techniques like DNN, CNN, LSTM, and RNN are compared with SRNN technique. Figure 3 represents the graphical representation of classification with FE using Buzzfeed. When compared to these existing techniques, SRNN achieves a high accuracy of 0.9847 due to the proposed approach leveraging sparse connectivity to enhance model efficiency and reduce overfitting, which allows it to better capture nuanced patterns in textual data.

Table 4 shows different feature extraction methods using PolitiFact. The existing techniques like BoW, Word2Vec, TD-IDF, and BERT are compared with WFT-BERT. Figure 4 represents a graphical representation of different feature extraction methods using PolitiFact. The WFT-BERT technique achieves a better accuracy of 0.9724 when compared to these existing techniques. WFT-BERT enhances performance by assigning varying significance to different segments or tokens which allows the model to focus more on appropriate parts of a text. This enhances contextual understanding and accuracy in fake news detection.

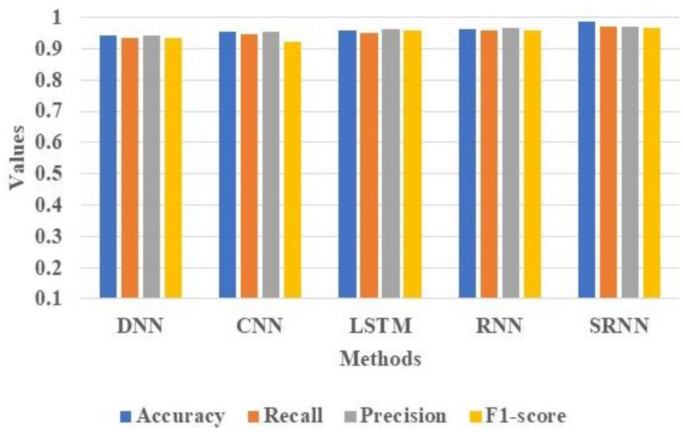

Figure 3. Graphical representation of classification performance utilizing Buzzfeed

Table 4. Different feature extraction methods utilizing PolitiFact dataset

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

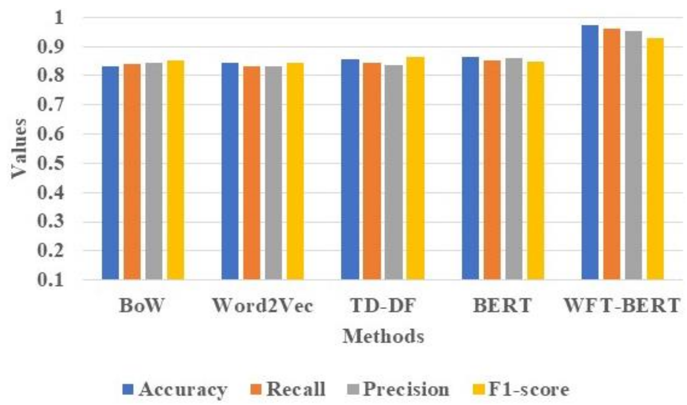

Figure 4. Graphical representation of different feature extraction methods utilizing PolitiFact

Table 5 shows the classification performances employing PolitiFact. The existing techniques like DNN, CNN, LSTM, and RNN are compared with the SRNN technique. Figure 5 represents the graphical representation of classification without FE using PolitiFact. SRNN achieves a superior accuracy of 0.9724 because the proposed approach leverages sparse connectivity by enhancing model efficiency compared to existing approaches like DNN, CNN, LSTM, and RNN.

Table 5. Classification performance utilizing PolitiFact dataset

Performance measures	DNN	CNN	LSTM	RNN	SRNN
Accuracy	0.9412	0.9456	0.9548	0.9625	0.9724
Recall	0.9354	0.9375	0.9456	0.9520	0.9612
Precision	0.9432	0.9357	0.9435	0.9468	0.9547
F1-score	0.9135	0.9257	0.9215	0.9265	0.9309

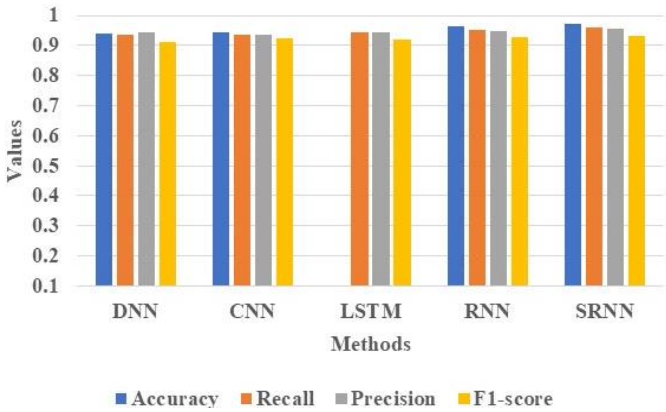

Figure 5. Graphical representation of classification performances using PolitiFact

4.2. Comparative analysis

Tables 6-8 display a comparative analysis between existing techniques and the proposed method on Buzzfeed, PolitiFact, Fakeddit, and Weibo datasets. The existing techniques like [17], [22] are exploited to compare with WFT-BERT-SRNN using Buzzfeed dataset. The studies [17], [20], [22] are the existing approaches used for comparison with the WFT-BERT-SRNN using Politifact. The studies [23], [24] are compared with the proposed method using Fakeddit and Weibo datasets. Compared to these existing techniques, the proposed WFT-BERT-SRNN achieves a better accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 for Buzzfeed, Politifact, Fakeddit, and Weibo datasets by leveraging BERT's contextual embeddings for rich text representation and SRNN's sequential modeling to capture patterns in news articles. This combination increases accuracy in distinguishing fake from genuine news by effectively managing temporal dependencies and complex linguistic features.

Table 6. Comparative analysis with existing techniques utilizing Buzzfeed

Performance measures	DeepFake [17]	EDU4FD [22]	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 7. Comparative analysis with existing techniques utilizing PolitiFact

Performance measures	DeepFake [17]	BERT-joint framework [20]	EDU4FD [22]	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 8. Comparative analysis with existing techniques using Fakeddit and Weibo datasets

Performance measures	Fakeddit dataset		Weibo dataset		
	Image caption-based technique [23]	Proposed WFT-BERT-SRNN	Image caption-based technique [23]	FMFN [24]	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

4.3. Discussion

This research demonstrates the WFT-BERT-SRNN to perform fake news detection. This approach achieves a notable enhancement in detection accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 for Buzzfeed, Politifact, Fakeddit, and Weibo datasets compared to existing methods. This improvement is attributed to the model's ability to leverage fine-tuned contextual embeddings and sparse recurrent processing to better capture the subtle patterns in fake news. The benefits of the proposed WFT-BERT-SRNN and limitations of existing techniques are discussed in this section. The limitations of existing techniques are: DeepFake [17] faces difficulties in evolving patterns and capturing dynamics in fake news because of its static nature. In SGCP+SVM [18], when a huge number of samples are lost, the presented technique decreases the detection performances. BERT-joint framework [20] faces difficulties in differentiating among various information types which affect the classification. EDU4FD [22] suffers from a potential loss of contextual coherence which leads to misinterpretation. Image caption-based technique [23] depends on a pre-defined vocabulary which struggles to accurately describe complex visual concepts. The proposed technique overcomes these existing technique's limitations.

In WFT-BERT, deeply bidirectional learning facilitates in achieving comparable or better outcomes than the conventional architecture in a particular task. SRNN increases the generalization by learning appropriate patterns and connections and gains efficiency and effectiveness in both inference and training. The primary purpose is to increase the accuracy and robustness of fake news detection systems by natural language processing. By incorporating fine-tuned BERT embedding with an SRNN, the proposed approach enhances the model's ability to discern subtle linguistic cues and contextual information associated with misinformation. This approach addresses the disadvantages of existing methods by integrating the strength of fine-tuned BERT embeddings with SRNN. The significance of the research lies in its potential to increase the reliability and accuracy of fake news detection systems.

5. CONCLUSION

Fake news detection is a key area of research because of wide spread impact of misinformation on public opinion, democratic processes, and social stability. This research proposed WFT-BERT-SRNN to detect fake news accurately. WFT-BERT enhances the model's ability to prioritize specific domains or tasks during training which allows it to capture specific information more effectively. This process improves the robustness and model's accuracy. SRNN lies in their ability to manage and learn large volumes of text data effectively by focusing on the most appropriate patterns and connections. SRNN increases generalization and enables the model to accurately classify news as fake or real. WFT-BERT-SRNN achieves a better accuracy of 0.9847, 0.9724, 0.9624, and 0.9725 for Buzzfeed, Politifact, Fakeddit, and Weibo datasets compared to existing methods like DeepFake, BERT-joint framework, EDU4FD, Image caption-based technique, and FMFN. These findings signify a substantial advancement in fake news detection field. The high accuracy indicates that WFT-BERT-SRNN acts as an appropriate tool for news agencies to identify and minimize the spread of misinformation. This approach provides a robust framework and increases the overall integrity of information. In the future, a hybrid DL detection method will be considered to enhance model performance.

FUNDING INFORMATION

The authors state no funding is involved.

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

Data availability does not apply to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] H. Saleh, A. Alharbi, and S. H. Alsamhi, "OPCNN-FAKE: optimized convolutional neural network for fake news detection," *IEEE Access*, vol. 9, pp. 129471–129489, 2021, doi: 10.1109/ACCESS.2021.3112806.

[2] D. K. Dixit, A. Bhagat, and D. Dangi, "Automating fake news detection using PPCA and levy flight-based LSTM," *Soft Computing*, vol. 26, no. 22, pp. 12545–12557, Jun. 2022, doi: 10.1007/s00500-022-07215-4.

[3] J. Jing, H. Wu, J. Sun, X. Fang, and H. Zhang, "Multimodal fake news detection via progressive fusion networks," *Information Processing and Management*, vol. 60, no. 1, p. 103120, Jan. 2023, doi: 10.1016/j.ipm.2022.103120.

[4] R. Sousa-Silva, “Fighting the fake: a forensic linguistic analysis to fake news detection,” *International Journal for the Semiotics of Law*, vol. 35, no. 6, pp. 2409–2433, Apr. 2022, doi: 10.1007/s11196-022-09901-w.

[5] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, “Fake news classification using transformer based enhanced LSTM and BERT,” *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, Jun. 2022, doi: 10.1016/j.ijcce.2022.03.003.

[6] J. Xu, V. Zadorozhny, D. Zhang, and J. Grant, “FaNDS: fake news detection system using energy flow,” *Data and Knowledge Engineering*, vol. 139, p. 101985, May 2022, doi: 10.1016/j.datak.2022.101985.

[7] A. Altheneyan and A. Alhadlaq, “Big data ML-based fake news detection using distributed learning,” *IEEE Access*, vol. 11, pp. 29447–29463, 2023, doi: 10.1109/ACCESS.2023.3260763.

[8] M. K. Jain, D. Gopalani, and Y. K. Meena, “ConFake: fake news identification using content-based features,” *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 8729–8755, Jun. 2024, doi: 10.1007/s11042-023-15792-1.

[9] S. Kumar, A. Kumar, A. Mallik, and R. R. Singh, “OptNet-Fake: fake news detection in socio-cyber platforms using grasshopper optimization and deep neural network,” *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4965–4974, Aug. 2024, doi: 10.1109/TCSS.2023.3246479.

[10] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, “A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media,” *IEEE Access*, vol. 10, pp. 62097–62109, 2022, doi: 10.1109/ACCESS.2022.3181184.

[11] C. Song, Y. Teng, Y. Zhu, S. Wei, and B. Wu, “Dynamic graph neural network for fake news detection,” *Neurocomputing*, vol. 505, pp. 362–374, Sep. 2022, doi: 10.1016/j.neucom.2022.07.057.

[12] L. Wu, Y. Rao, C. Zhang, Y. Zhao, and A. Nazir, “Category-controlled encoder-decoder for fake news detection,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1242–1257, 2023, doi: 10.1109/TKDE.2021.3103833.

[13] S. Rezaei, M. Kahani, B. Behkamal, and A. Jalayer, “Early multi-class ensemble-based fake news detection using content features,” *Social Network Analysis and Mining*, vol. 13, no. 1, Dec. 2023, doi: 10.1007/s13278-022-01019-y.

[14] M. C. de Souza, B. M. Nogueira, R. G. Rossi, R. M. Marcacini, B. N. dos Santos, and S. O. Rezende, “A network-based positive and unlabeled learning approach for fake news detection,” *Machine Learning*, vol. 111, no. 10, pp. 3549–3592, Nov. 2022, doi: 10.1007/s10994-021-06111-6.

[15] M. Albahar, “A hybrid model for fake news detection: Leveraging news content and user comments in fake news,” *IET Information Security*, vol. 15, no. 2, pp. 169–177, Mar. 2021, doi: 10.1049/ise2.12021.

[16] S. Senhadji and R. A. S. Ahmed, “Fake news detection using naïve Bayes and long short-term memory algorithms,” *IAES International Journal of Artificial Intelligence*, vol. 11, no. 2, pp. 746–752, Jun. 2022, doi: 10.11591/ijai.v11.i2.pp746-752.

[17] R. K. Kaliyar, A. Goswami, and P. Narang, “DeepFakE: improving fake news detection using tensor decomposition-based deep neural network,” *Journal of Supercomputing*, vol. 77, no. 2, pp. 1015–1037, May 2021, doi: 10.1007/s11227-020-03294-y.

[18] H. Che, B. Pan, M. F. Leung, Y. Cao, and Z. Yan, “Tensor factorization with sparse and graph regularization for fake news detection on social networks,” *IEEE Transactions on Computational Social Systems*, vol. 11, no. 4, pp. 4888–4898, Aug. 2024, doi: 10.1109/TCSS.2023.3296479.

[19] B. Palani, S. Elango, and K. V. Viswanathan, “CB-Fake: A multimodal deep learning framework for automatic fake news detection using capsule neural network and BERT,” *Multimedia Tools and Applications*, vol. 81, no. 4, pp. 5587–5620, Dec. 2022, doi: 10.1007/s11042-021-11782-3.

[20] W. Shishah, “Fake news detection using BERT model with joint learning,” *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 9115–9127, Jun. 2021, doi: 10.1007/s13369-021-05780-8.

[21] R. K. Kaliyar, A. Goswami, and P. Narang, “EchoFakeD: improving fake news detection in social media with an efficient deep neural network,” *Neural Computing and Applications*, vol. 33, no. 14, pp. 8597–8613, Jan. 2021, doi: 10.1007/s00521-020-05611-1.

[22] Y. Wang, L. Wang, Y. Yang, and Y. Zhang, “Detecting fake news by enhanced text representation with multi-EDU-structure awareness,” *Expert Systems with Applications*, vol. 206, p. 117781, Nov. 2022, doi: 10.1016/j.eswa.2022.117781.

[23] P. Liu, W. Qian, D. Xu, B. Ren, and J. Cao, “Multi- modal fake news detection via bridging the gap between modals,” *Entropy*, vol. 25, no. 4, p. 614, Apr. 2023, doi: 10.3390/e25040614.

[24] J. Wang, H. Mao, and H. Li, “FMFN: fine-grained multimodal fusion networks for fake news detection,” *Applied Sciences (Switzerland)*, vol. 12, no. 3, p. 1093, Jan. 2022, doi: 10.3390/app12031093.

[25] BuzzFeed News Dataset (2018–2023), Kaggle. [Online]. Available: https://www.kaggle.com/datasets/konradb/buzzfeed-news-2018-2023. Accessed: Jan. 2026.

[26] PolitiFact Fake News Dataset, Kaggle. [Online]. Available: https://www.kaggle.com/datasets/subodh7300/politifact/code. Accessed: Jan. 2026.

[27] Fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection, GitHub. [Online]. Available: https://github.com/entitize/Fakeddit. Accessed: Jan. 2026.

[28] Weibo public opinion datasets, GitHub. [Online]. Available: https://github.com/hlbao/weibo-public-opinion-datasets. Accessed: Jan. 2026.

[29] N. N. Prachi, M. Habibullah, M. E. H. Rafi, E. Alam, and R. Khan, “Detection of fake news using machine learning and natural language processing algorithms,” *Journal of Advances in Information Technology*, vol. 13, no. 6, pp. 652–661, 2022, doi: 10.12720/jait.13.6.652-661.

[30] M. Yarlagadda, K. Gangadhara Rao, and A. Srikrishna, “Frequent itemset-based feature selection and rider moth search algorithm for document clustering,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1098–1109, Apr. 2022, doi: 10.1016/j.jksuci.2019.09.002.

[31] D. Viji and S. Revathy, “A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi – LSTM model for semantic text similarity identification,” *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6131–6157, Jan. 2022, doi: 10.1007/s11042-021-11771-6.

[32] M. V. Dasu, P. V. N. Reddy, and S. C. M. Reddy, “Deep concatenated features with improved heuristic-based recurrent neural network for hyperspectral image classification,” *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 49875–49904, Nov. 2024, doi: 10.1007/s11042-023-17351-0.

BIOGRAPHIES OF AUTHORS

Asha Kathigi is working as an Assistant Professor in the department of artificial intelligence and machine learning. She has 17 years of experience in teaching. She is teaching subjects like DAA, Python programming, microcontrollers, Java programming, management and entrepreneurship, and computer organization. Her area of interest is image processing and pattern recognition. Currently, she is doing research on pattern recognition. She can be contacted at email: info@gmit.ac.in.

Gautam Vinod Nair is fervently engaged in web development and machine learning pursuits. With a specialized focus on artificial intelligence and machine learning from GM Institute of Technology, he has honed his craft at Sookshmas E-Learning Pvt Ltd, where he curated visually engaging interfaces and managed databases. His roles at Compsoft Technologies and Bharat Intern enriched his understanding of machine learning methodologies, encompassing data preprocessing and model optimization. Notable projects like facial recognition using MATLAB and wine quality prediction through computational analysis underscore his commitment to innovation. Beyond technology, he indulges in basketball and exploration. Recognition as a Microsoft Learn Student Ambassador underscores his dedication. He can be contacted at email: 4gm20ai013@gmit.ac.in.

Kruthika Kadurahalli Raghu is currently pursuing her bachelor of technology in artificial intelligence and machine learning at GM Institute of Technology, Davangere. Throughout her engineering journey, she has actively engaged in various projects and internships, focusing on honing her skills in Python programming, data analysis, and web development. From creating a hotel management system to detecting skin diseases using image processing, she has explored the diverse applications of technology. Beyond academics, she is deeply passionate about continuous learning, evident in her participation in workshops and certifications, including those from Google and Infosys. She also actively contributes to co-curricular and extracurricular activities, from organizing technical fests to participating in sports and community service initiatives like blood donation drives. She can be contacted at email: kruthikakr45@gmail.com.

Kavya Pujar Prakash is currently embarking on her journey towards a Bachelor's degree in Artificial Intelligence and Machine Learning at GM Institute of Technology. With a fervent enthusiasm for deciphering data and unraveling the mysteries of machine learning, she have dived deep into the realms of Python, Java, C++, and an array of data analysis tools. Her stint at Technologics Global Pvt. Ltd, coupled with ongoing projects, has furnished her with hands-on experience in deploying AI and ML techniques to tackle real-world conundrums like brain tumor detection and soil pH estimation. Certified in Java and Artificial Intelligence, her accolades extend to active participation in events like the Smart India Hackathon and completion of enriching workshops on Machine Learning and Robotics. Upholding an unyielding commitment to perpetual learning, she is driven by an unwavering determination to excel and innovate in the dynamic domain of artificial intelligence and machine learning. She can be contacted at email: kavyapujar16@gmail.com.

Meghana Deepak Duttargi is fueled by a passion for Machine Learning and a knack for data analysis. Currently pursuing a B.E. in artificial intelligence and machine learning at GM Institute of Technology, she thrives on unraveling complex data puzzles and crafting innovative solutions. With hands-on experience in Python, MySQL, and AI techniques, garnered through internships at Technologics Global Pvt. Ltd and Technohack, she have delved into projects ranging from brain tumor detection to wine quality prediction.
Outside academia, she has completed internships in full-stack development and actively participated in workshops on machine learning and robotics. Certified by Deeplearning. AI and Google, she is dedicated to staying at the forefront of technological advancements. Whether it's developing a Hotel Management System or exploring skin disease detection, she is always eager to tackle new challenges head-on. She can be contacted at email: meghanaduttargi26129@gmail.com.