An Approach Based on Fine-tuning Small Language Models for Fake News Detection

Original Scientific Paper

Khac-Lap Phan
Department of Information Technology, Quy Nhon University
170 An Duong Vuong, Quy Nhon Nam, Gia Lai, Vietnam
lap4654100006@st.qnu.edu.vn

Quang-Vinh Pham
Department of Data Science, TMA Tech Group
Quy Nhon Nam, Gia Lai, Vietnam
pqvinh@tma.com.vn

Quang-Hung Le*
Department of Information Technology, Quy Nhon University
170 An Duong Vuong, Quy Nhon Nam, Gia Lai, Vietnam
lequanghung@qnu.edu.vn

*Corresponding author

Abstract *– Social networks have facilitated the rapid spread of fake news. This can negatively affect the society. Therefore, effective fake news detection is growing increasingly important. Recently, large language models (LLMs) have brought out impressive results. However, practical challenges remain because LLMs require significant computational resources. This paper explores the potential of high-performance small language models (SLMs) as an effective alternative in edge and low-resource settings. Particularly, we propose SLM-FND, a lightweight yet effective framework based on fine-tuning small language models. We evaluated state-of-the-art models, including DistilBERT, MiniLM, and ALBERT using both full fine-tuning and parameter-efficient fine-tuning methods on benchmark datasets. The results show that the proposed method achieves approximately 98.3% accuracy and F1-score values on balanced datasets. This indicates that SLMs yield competitive performance to LLM-based baselines, while requiring significantly fewer trainable parameters, shorter inference times, and lower hardware cost. Furthermore, parameter-efficient fine-tuning methods promote adaptability and minimize the computational cost. The code for the implementation of this work has been shared in the following GitHub repository (https://github.com/PhanKhacLapQNU/SLM-FND).*

Keywords: *Fake News Detection, Small Language Models, Parameter-Efficient Fine-Tuning, SLM-FND*

Received: February 1, 2026; Received in revised form: May 11, 2026; Accepted: May 11, 2026

1. INTRODUCTION

In the current digital landscape, fake news and misleading information can cause more damage than actual viruses, posing a massive threat to social cohesion and stability [1]. This threat has been highlighted during the COVID-19 pandemic, when inaccurate vaccine-related misinformation led to ill-informed decisions and harmful health effects [2]. As misleading content proliferates on TikTok, Facebook, and Twitter [3], developing machine learning and natural language processing tools for fake news detection (FND) has emerged as a key research priority [4].

Although LLMs including GPT-4 [5] and BERT [6] achieve strong predictive performance, they incur considerable computational overhead, including high inference latency and substantial memory demands. These limitations restrict their practical deployment on edge devices and resource-limited servers, especially in environments with uneven technological infrastructure [5, 6]. This motivates the development of SLMs that aim to balance model effectiveness with markedly reduced resource consumption [7, 8]. This study is focused on the fine-tuning of SLMs for FND and provides a systematic analysis of the trade-offs between predictive accuracy and computational efficiency. To the best of our knowledge, this is the first study to conduct a comprehensive exploration of the state-of-the-art SLMs, including DistilBERT [8], MiniLM [9], and ALBERT [10] using both full fine-tuning and parameter-efficient

fine-tuning (PEFT) methods. Our main contributions are summarized as follows:

1. We propose SLM-FND, a lightweight yet effective framework based on fine-tuning SLMs for detecting fake news.
2. We investigated the potential of PEFT methods to reduce the computational cost while maintaining a competitive model performance.
3. We conducted a systematic analysis of the SLM-based approach and baselines, providing practical insights into the trade-offs in terms of detection accuracy, computational efficiency, and deployment feasibility.

The remainder of this paper is organized as follows. The mathematical foundations, background, and related work are reviewed in Section 2. Section 3 details our methodology, including models and fine-tuning strategies, training procedure, and classification layer. Section 4 describes the experimental setup. Section 5 analyzes and discusses the obtained evaluation results. Finally, Section 6 concludes and outlines future research directions.

2. BACKGROUND AND RELATED WORK

In this section, we first provide the mathematical foundations and briefly present SLMs. Then, we discuss some related work on FND techniques.

2.1. TASK FORMULATION

The FND task can be formulated as follows. Let X be the input space of all possible textual sequences and Y={0,1} be the binary label space. We define $D=\{(x_i, y_i)\}_{i=1}^{N}$ to denote a labeled dataset of news articles, headlines, or short text segments, where:

- D represents the complete collection of available data instances,
- $x_i \in X$ represents a specific textual instance (e.g., a full article text, a headline-body pair, or a social media post),
- $y_i \in Y$ denotes the corresponding label, with 0 indicating real news and 1 indicating fake news,
- N is the total number of examples in the dataset D.

The objective is to learn a mapping function

$$f_\theta: X \rightarrow Y \quad (1)$$

parameterized by θ (which denotes the learnable weights of the model), such that the predicted labels $\hat{y}_i = f_\theta(x_i)$ approximate the true labels y_i as closely as possible over the dataset D.

2.2. SMALL LANGUAGE MODELS (SLMS)

As discussed earlier, SLMs have recently emerged as a potential alternative to LLMs owing to their low inference latency and cost-effectiveness. SLMs are designed to retain strong language understanding capabilities while significantly reducing the computational costs. These compact architectures are generally derived from larger models using three primary compression techniques: *(i) knowledge distillation*, where a student model learns to mimic the behavior of a teacher [7]; *(ii) pruning*, which removes less critical weights to reduce size [12]; and *(iii) quantization*, which lowers the parameter precision to minimize memory footprint.

2.3. FAKE NEWS DETECTION TECHNIQUES

The evolution of the FND task has progressed from manual feature generation to deeper semantic analysis [13-15]. Early approaches relied primarily on traditional ML algorithms such as Logistic Regression (LR), SVM, or Random Forest [16, 17]. For example, Ahmed et al. [18] utilized N-gram analysis alongside classifiers like SVM to detect online fake news. While computationally efficient, these surface-level models struggle to capture complex semantic dependencies.

Subsequently, deep learning architectures such as CNNs and LSTMs were introduced to automatically extract localized and sequential text patterns [19-22]. Kim [23] pioneered the application of CNNs for sentence classification by utilizing multiple filter sizes to capture local semantic features without manual engineering. Concurrently, for sequential modeling, Ruchansky et al. [24] introduced the CSI framework, leveraging LSTM networks to integrate the temporal user behavior with text representations. Although these neural networks improve detection accuracy, their architectural constraints, including fixed filter windows and sequential bottlenecks, limit their ability to comprehend long-range linguistic contexts.

Recently, pre-trained transformer models have emerged, demonstrating capabilities in understanding complex semantics [25, 26], with BERT and RoBERTa being considered the most advanced examples to date [6, 11]. To resolve long-range dependency issues, Kaliyar et al. [27] developed FakeBERT, a hybrid architecture combining a robust BERT encoder with CNN classifiers, demonstrating that bidirectional transformers significantly outperform traditional deep learning models. However, the increasing parameter size of these pre-trained models inherently results in high computational demands, driving the field toward larger architectures.

The most recent turn in FND research has been facilitated by the integration of LLMs to tackle complex reasoning and data constraints. For instance, Hu et al. [28] utilized LLMs as multi-perspective rationale generators to guide smaller classifiers. Other studies leveraged LLM-driven data augmentation to mitigate dataset imbalance [29] and cross-lingual limitations [30], while Yang et al. [31] applied LLM-based user modeling to handle unseen domain shifts. Although these strategies significantly improve generalization, their reliance

on continuous API calls, synthetic data generation, and multi-stage training pipelines introduces high latency and operational complexity.

Parallel advancements have focused on enriching contextual understanding through multimodal and robust architectures. Wang et al. [32] and Habib et al. [33] developed LLM-enhanced frameworks to dynamically fuse textual, visual, and temporal signals, whereas Bharati et al. [34] modeled multimodal uncertainty using fuzzy logic. Additionally, to counteract the emerging problem of sentiment manipulation using LLM, Tahmasebi et al. [35] proposed adversarial training pipelines. However, these models demand extensive computational resources and massive parameter scales.

Despite the aforementioned advances, there still lacks a systematic study about whether such models may be applicable in real-world environments, especially in edge computing and low-resource settings. This gap motivated us to explore the relationship between detection performance and system-specific limitations such as memory, inference speed, and computational resources in the context of misinformation detection.

This paper addresses the following research questions (RQs):

- RQ1: How well can SLMs detect fake news as compared to larger pre-trained language models?
- RQ2: How do PEFT methods influence SLM performance?
- RQ3: What are the trade-offs between accuracy and computational efficiency when the resources are constrained?

3. METHODOLOGY

This section presents the proposed framework (namely, SLM-FND) based on fine-tuning SLMs for detecting fake news. As illustrated in Fig. 1, our approach utilized a systematic, stepwise processing flow. First, we cleaned all the pertinent data, normalized the news articles, and tokenized them to meet the input requirement for transformer-based algorithms. Second, these tokenized sequences were sent to the SLM backbone, which captured their contextual information and encoded it into meaningful representations. Finally, these representations are input to a linear classifier, which uses them linearly to output the model prediction.

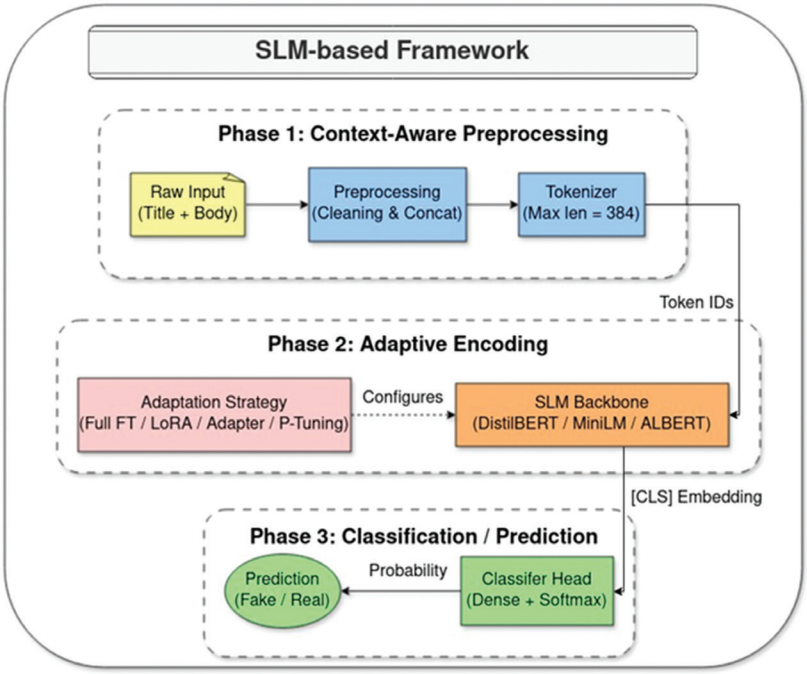


Fig. 1: Overview of the proposed SLM-FND framework

3.1. MODELS AND FINE-TUNING STRATEGIES

In this work, we use three SLMs: DistilBERT, MiniLM, and ALBERT. These models apply different architectural strategies for efficient language representation. DistilBERT [8] adopts knowledge distillation to shrink a BERT-sized model while largely preserving its representational capacity. MiniLM [9] distills self-attention behaviors from larger architectures, yielding a lightweight model with roughly 33 million parameters, which is suitable for edge computing and low-resource settings. ALBERT [10] enhances memory efficiency by sharing parameters across layers and factorizing embeddings, resulting in a highly compact architecture with roughly 11 million parameters. Taken together, these models enable an empirical analysis of how different compression strategies influence contextual representation quality and overall performance when compared against standard large-scale baselines such as BERT and RoBERTa.

This study examined two principal strategies for adapting these backbone models to the binary classification task: full fine-tuning and PEFT. Under the conventional full fine-tuning paradigm, all model parameters θ are optimized via backpropagation throughout the network. Although this approach is effective, it entails substantial storage overhead because a complete model instance must be maintained for each downstream task. To address this limitation, we adopted PEFT, which restricts optimization to a small subset of parameters, and implement three representative variants encompassing weight-, module-, and input-level adaptation mechanisms.

The first PEFT strategy is low-rank adaptation (LoRA) [36], which injects trainable low-rank decomposition matrices into the attention layers, specifically the query and value matrices, while freezing the pretrained weights $W_0 \in R^{d \times k}$. The forward pass is modified as:

$$h = W_0 x + \frac{\alpha}{r} BAx \tag{2}$$

where, x denotes the input vector, h is the output vector, $B \in R^{d \times k}$ and $A \in R^{r \times k}$ are low-rank matrices, r represents the rank of the decomposition ($r \ll min^{d,k}$), and α denotes a constant scaling factor. Alternatively, we utilized bottleneck adapters [37], which insert lightweight modules within each transformer block after the attention and feed-forward layers. Defined by a down-projection W_{down} and up-projection W_{up}, the adapter output is computed as:

$$Adapter(h) = W_{up} \cdot \sigma(W_{down} \cdot h) + h \tag{3}$$

where, σ represents a non-linear activation function (such as ReLU or GELU), and only the adapter parameters and layer normalization are trained. Finally, we explore prompt tuning [38], which optimizes the input space rather than the model architecture. By prepending m trainable soft prompt vectors P to the input embeddings E of sequence length n, the augmented input becomes:

$$X' = [P; E] = [p_1; \ldots; p_m; e_1; \ldots; e_n] \tag{4}$$

where, p_i and e_i denote the individual prompt and input token embeddings, respectively. This strategy freezes the entire pre-trained backbone, representing the most memory-efficient approach in our study.

3.2. TRAINING PROCEDURE

The fine-tuning procedure was uniformly applied across all models and datasets to guarantee a fair and consistent comparison, with training guided by the objective of minimizing empirical risk on the training data. Model optimization was performed using the cross-entropy loss, which quantifies the divergence between the predicted probability distribution and the ground-truth labels. In particular, a weighted variant of the cross-entropy loss was adopted:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} w_{y_i} \log(p(y_i|x_i)) \tag{5}$$

Here, N represents the number of samples in a given training batch, x_i is the i-th input sequence, $p(y_i|x_i)$ denotes the probability predicted by the model for the true class y_i, and w_{yi} denotes the inverse-frequency weight associated with the class y_i. w_{yi} was introduced to explicitly address the potential class imbalance within the datasets. Furthermore, we used the AdamW optimizer with a decoupled weight decay set as 0.01 to mitigate overfitting, together with a linear learning rate schedule that incorporates a warmup phase corresponding to 10% of the total training steps. For full fine-tuning, the learning rate was initialized at 2×10^{-5}, while PEFT-based approaches employed a larger initial rate of 1×10^{-3}, agreeing with the higher optimization sensitivity commonly observed in parameter-efficient methods. Model training was performed using batch sizes of 16 or 32, depending on the availability of GPU memory, for a total of 3 to 5 epochs. The dropout probability was fixed at 0.1. An early stopping strategy guided by the validation macro-$F1$-score (patience of three epochs) was employed to select the optimal model checkpoint.

3.3 CLASSIFICATION LAYER

The classification head was appended to the final layer of the SLM backbone, which extracted the hidden state of the special classification token ([CLS]) from the last transformer layer, denoted as $h_{[CLS]} \in R^d$. This vector served as the aggregate representation of the entire news article. It was passed through a fully-connected dense layer followed by a Softmax activation function to produce the probability distribution over the two classes (*fake* and *real*) as:

$$\hat{y} = Softmax(W_{cls} \cdot h_{[CLS]} + b_{cls}) \tag{6}$$

where, $W_{cls} \in R^{2 \times d}$ and $b_{cls} \in R^2$ are the trainable parameters of the classifier. The complete training procedure, encompassing data standardization and strategy selection, is summarized in Algorithm 1.

Algorithm 1 General training framework for SLM-based FND

Require: Dataset D WELFake, FakeNewsNet, LIAR; Backbone M; Strategy S; Hyperparams η, B, E.

Ensure: Optimized Parameters θ^*.

// **Phase 1:** Context-Aware Preprocessing

1: **for** each sample $(x_i, y_i) \in D$ **do**

2: **if** Dataset is **LIAR then**

3: *Fusion*: $raw_{text} \leftarrow Statement \oplus [SEP] \oplus Context \oplus [SEP] \oplus Speaker$

4: **else** ▷ WELFake / FakeNewsNet

5: Concat: $raw_{text} \leftarrow Title \oplus [SEP] \oplus BodyText$

BodyText

6: **end if**

7: $x_i \leftarrow Clean(raw_{text})$; $T_i \leftarrow Tokenize(x_i)$

8: **end for**

// **Phase 2:** Model Configuration (PEFT Selection)

9: Load pre-trained weights θ.

10: **if** $S == LoRA$ **then**

11: Inject matrices A, B into Attention; Freeze θ.

12: **else** if θ **then**

13: Insert Bottleneck Layers; Freeze θ.

14: **else if** $S ==$ PromptTuning **then**

15: Prepend Soft Prompts P; Freeze θ.

16: **else**

17: Unfreeze all parameters full fine-tuning.

18: **end if**

// **Phase 3:** Training

19: Optimize θ_{train} using Weighted Cross-Entropy.

20: **return** θ^*.

4. EXPERIMENTAL SETUP

The experiments aimed to answer the aforementioned three research questions (RQ1 to RQ3). This section describes the datasets, baselines, evaluation metrics, and implementation specifics.

4.1. DATASETS

To evaluate how the proposed method performs across various scales and domains, we utilized three benchmark datasets, namely, WELFake [39], FakeNewsNet [40], and LIAR [41]. Table 1 shows the useful statistics after we carefully processed the data.

- WELFake: This dataset is a large-scale corpus of general-domain news articles. In particular, 72,134 samples were originally collected, and 63,323 samples were retained after the application of preprocessing procedures intended to ensure high-quality text. Given its near-balanced class distribution, with 54.5% of samples labeled as *fake* and 45.5% as *real*, only limited class reweighting was required during training, with the corresponding weight for the fake class set to $w_{fake} \approx 0.92$.
- FakeNewsNet: This corpus gathered the news shared on social media. From an original sample size of 23,196 entries, FakeNewsNet was filtered down to 21,287 valid samples by removing those without proper text. The final split is heavily skewed, as fake news represents just 24.2% of the data, thereby demanding fixes during model training. In particular, we implemented inverse frequency-based class reweighting, setting weights to approximately $w_{fake} \approx 2.06$ and $w_{real} \approx 0.66$.
- LIAR: The LIAR dataset is widely used for evaluating short-text fact-checking models. Its original six-category labeling scheme was reformulated into a binary classification problem (*fake versus real*), producing a total of 22,962 samples. The dataset was split into 18,369 training samples, 2,296 validation samples, and 2,297 test samples. The class distribution is not highly uneven, with 39.4% fake samples and 60.6% real samples. This problem was fixed by adding class weights to the loss function ($w_{fake} \approx 1.27$, $w_{real} \approx 0.82$). In addition, because the text inputs are short, metadata was used to provide more context.

Table 1. Dataset statistics after pre-processing

Dataset	Effective Samples	Train / Val / Test	Balance (Fake : Real)	Class Weights
WELFake	63,323	47.5k / 7.9k / 7.9k	54.5 : 45.5	0.92 : 1.10
FakeNewsNet	21,287	16.0k / 2.1k / 3.2k	24.2 : 75.8	2.06 : 0.66
LIAR	22,962	18.4k / 2.3k / 2.3k	39.4 : 60.6	1.27 : 0.82

4.2. BASELINES

To establish comparative performance, we included two baseline categories:

1. Traditional machine/deep learning methods. We employed LR on the top 50,000 n-grams using TF-IDF features. For high-dimensional predictive power, we used SVM with LinearSVC and squared hinge loss. For a deep learning baseline, we utilized bidirectional LSTMs (Bi-LSTM), which started with pretrained word embeddings to capture sequential dependencies.

2. Pre-trained models-based methods. We employed full-scale transformer models, including the standard BERT-base-uncased (110M parameters) [6], RoBERTa-base (125M parameters) [11] and RoBERTa-large (355M parameters [11]. The RoBERTa models are highly optimized versions that often achieve the best results.

4.3. EVALUATION METRICS

We use accuracy (Acc), precision (P), recall (R), and $F1$-score ($F1$) to evaluate categorical predictions:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN} \tag{8}$$

$$F_1 = 2 \times \frac{P \times R}{P+R} \tag{9}$$

where, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

We formally define computational efficiency using two key indicators:

- Inference throughput (τ): Defined as the number of samples processed per unit time:

$$\tau = \frac{B \times K}{t_{batch}} (samples/second) \tag{10}$$

where, B denotes the batch size, K is the number of batches, and tbatch denotes the average processing time per batch on the GPU.

- Training cost (T_{train}): The total wall-clock time required for model convergence:

$$T_{train} = \sum_{e=1}^{E} t_{epoch}^{(e)} \quad (11)$$

where, E denotes the total number of epochs and $t_{epoch}^{(e)}$ represents the duration of the e^{th} epoch.

4.4. IMPLEMENTATION DETAILS

Experiments were conducted on a Tesla T4 GPU (16GB VRAM) using PyTorch 2.0 and the Hugging Face PEFT library [42]. Hyperparameters were optimized via grid search, as summarized in Table 2.

Table 2. Summary of training configurations and hyperparameters

Strategy	*LR*	*BS*	*E*	Parameters
Full FT	2×10^{-5}	16	3	*WD*: 0.01
LoRA [36]	1×10^{-3}	32	5	*r*, *α*, *Dr*
Adapters [37]	1×10^{-3}	32	5	*BR*
Prompt Tuning [38]	1×10^{-2}	32	5	8*T*

Note: FT: fine-tuning, *LR*: learning rate, *BS*: batch size, *E*: epochs, *WD*: weight decay, *BR*: bottleneck reduction, *T*: virtual tokens, *Dr*: dropout rate (0.1), *r*: LoRA rank (16), *α*: LoRA scaling factor (32)

5. RESULTS AND DISCUSSION

5.1. RESULTS

Table 3 provides an overview of accuracy and $F1$-score metrics across the three benchmark datasets [39-41]. In particular, SLMs achieve a performance similar to that of their corresponding teachers on balanced datasets like WELFake, with DistilBERT and MiniLM, achieving an $F1$-score of 98.3%. However, a significant performance drop was observed on unbalanced (FakeNewsNet) and noisy (LIAR) data, where prompt tuning performed considerably less effectively, dropping to 65.4% and 45.7% $F1$-scores, respectively. Conversely, the bottleneck adapters maintained the highest stability on short text statements, achieving an $F1$-score of 67.8%.

As mentioned in Table 4, adding additional prompt tokens stops being useful above a certain limit. This limit is approximately 32 tokens, at which most models approach saturation. If we increase it to 64, the additional context only becomes redundant, causing the semantic dilution and performance of DistilBERT and MiniLM to stagnate and even decrease, especially in noisy datasets such as LIAR. However, ALBERT proves to be an outlier in this case, only because it is the only architecture that continues to improve as the prompt grows longer. This indicates that the performance of prompt tuning depends heavily on how the specific model handles context. No single size can optimize all models. This is an obvious trade-off that needs to be considered for every model and dataset considered in this study.

Table 5 describes the performance of these models in terms of storage and cost and their performance on different datasets to help us understand their overall costs in terms of storage, GFLOPs [43] and inference speed. Gaining a glimpse of the efficiency landscape and accuracy does not suffice if the model is too heavy to be implemented. These are the actual trade-offs involved in practical application of these architectures.

Table 3. Consolidated performance metrics ($F1$-score and accuracy) of all models considered in this study across three bench-mark datasets.

Method	WELFake		FakeNewsNet		LIAR	
	Acc	*F*1	Acc	*F*1	Acc	*F*1
Baselines						
Logistic Regression [16]	96.1	96.1	84.3	83.9	60.5	60.9
LinearSVC [44]	97.7	97.7	84.3	83.9	62.4	57.2
Bi-LSTM [19]	97.9	97.9	77.9	78.9	59.6	59.1
BERT (Full) [6]	98.7	98.7	83.6	83.6	63.7	63.1
RoBERTa Base (Full) [11]	99.4	99.4	**84.6**	**85.0**	63.1	63.0
RoBERTa Large (Full) [11]	**99.5**	**99.5**	75.8	65.3	60.6	45.7
Ours (SLMs)						
DistilBERT (Full)	98.3	98.3	83.8	84.1	63.6	63.2
DistilBERT (LoRA)	96.2	96.2	82.7	82.9	58.3	58.5
DistilBERT (Adapt)	97.0	97.0	81.4	81.8	**68.4**	**67.8**
DistilBERT (Prompt-8)	96.8	96.8	75.8	65.4	60.0	45.7
MiniLM (Full)	98.3	98.3	80.6	81.2	63.5	63.3
MiniLM (LoRA)	95.0	95.0	81.1	81.7	60.6	45.7
MiniLM (Adapt)	96.3	96.3	79.0	79.9	67.4	66.5
MiniLM (Prompt-8)	94.8	94.8	73.7	74.9	58.8	55.7
ALBERT (Full)	97.7	97.7	81.4	81.9	60.6	45.7
ALBERT (LoRA)	95.6	95.6	75.8	65.3	60.6	45.7
ALBERT (Adapt)	97.4	97.4	77.5	78.6	**68.6**	**67.8**
ALBERT (Prompt-8)	96.2	96.2	76.1	77.1	57.2	57.6

Table 4. Comparative performance ($F1$-score) of DistilBERT, MiniLM, and ALBERT across different prompt lengths and datasets

Dataset	Model	F_1 (8 tks)	F_1 (32 tks)	F_1 (64 tks)	Trend
WELFake	DistilBERT	96.8	96.90	97.05	Slight Increase
	MiniLM	94.8	95.75	95.89	Slight Increase
	ALBERT	96.2	96.97	97.03	Slight Increase
FakeNewsNet	DistilBERT	65.4	65.39	65.39	Saturation
	MiniLM	74.9	74.85	74.73	Peak at 8
	ALBERT	77.1	79.27	79.04	Peak at 32
LIAR	DistilBERT	45.7	45.71	45.71	Collapse
	MiniLM	55.7	50.69	45.71	Peak at 8
	ALBERT	57.6	57.61	58.11	Gradual Increase

Table 5. Consolidated efficiency profile of different models across three benchmark datasets when inference was achieved through full fine-tuning

Backbone	Storage		Cost	WELFake (Long)		FakeNewsNet		LIAR (Short)	
	Params	Size	GFLOPs	Smp/s	Lat(ms)	Smp/s	Lat(ms)	Smp/s	Lat(ms)
RoBERTa-large	355M	≈1.4GB	80	52	19.20	257	3.89	88	11.35
RoBERTa-base	125M	≈500MB	22.5	187	5.35	675	1.48	477	2.10
BERT-base	110M	≈440MB	22.5	178	5.62	1300	0.77	638	1.57
DistilBERT	66M	268MB	11.3	285	3.51	1045	0.96	1276	0.78
MiniLM	33M	≈134MB	10.8	**519**	**1.93**	**3211**	**0.31**	**862**	**1.16**
ALBERT-base	**11M**	**≈47MB**	22.5	94	10.64	1237	0.81	560	1.79

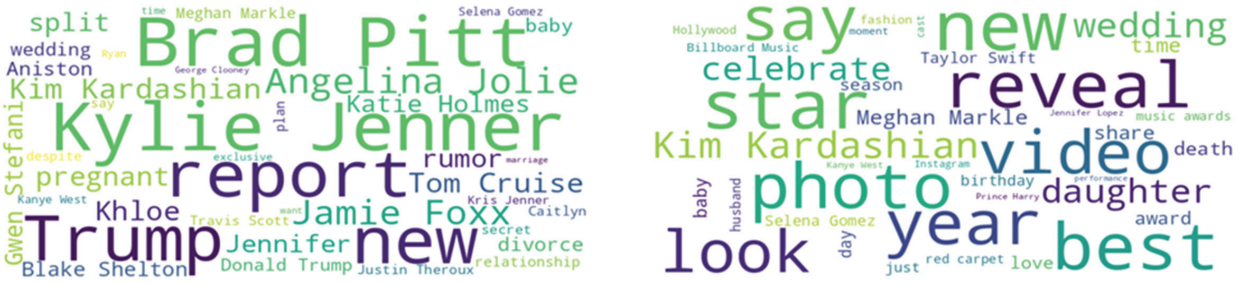
Note: Parameters: Number of parameters (millions); Size: Model size on disk (MB); Smp/s: Samples/second; Lat(ms): Latency (milliseconds)

In terms of resource utilization (in Table 5), MiniLM achieved the highest throughput in all scenarios, with up to 3211 samples/s of short text and a latency of 0.31 ms. On the other hand, ALBERT achieved the smallest model size (≈47MB), but produced a significantly higher latency (10.64 ms) compared with the SLMs within long-text scenarios. Based on these objective observations, we provide a qualitative trade-off analysis in Section 5.3.

5.2. ERROR ANALYSIS

To understand the linguistic patterns causing misclassification, we performed a quantitative keyword frequency analysis on the prediction logs of the model to identify specific trends. This audit examines 37 False Positive (FP) and 24 False Negative (FN) samples from the WELFake dataset to understand the error patterns of the model during testing.

- False Positives (Fig. 2a): Quantitative results reveal that 51.35% of FP errors were driven by high-profile political and celebrity entities (e.g., "*Trump*", "*Brad Pitt*", "*Kylie Jenner*"). This confirms that the model tends to conflate tabloid-style sensationalism with misinformation. Furthermore, 16.22% of FPs were attributed to a "bureaucratic mimicry" strategy, where authoritative terms like "*Treasury*", "*Directive*", or "*Official*" mislead the model into a real classification.
- False Negatives (Fig. 2b): Fake news that evades detection often utilizes a benign context strategy. Statistical analysis shows that 20.83% of FN cases utilized positive sentiment tokens (e.g., "*wedding*", "*love*", "*celebrate*") and entertainment references like "*Taylor Swift*". This indicates that fabricated content masquerading as harmless social updates lacks the inflammatory language typically associated with fake news, allowing it to bypass the classifier.



(a) (b)

Fig. 2. Word cloud analysis of model errors: visualizing the most frequent words found in (**a**) false positive and (**b**) false negative predictions

Table 6 presents specific qualitative examples of prediction errors. As discussed earlier, the samples of false positives (rows 1-2) show that the model appears deceived by a "*bureaucratic mimicry*" strategy, where the presence of authoritative entities such as "*Treasury Department*" or "*Archives*" overrides the falsity of the claim, leading to an incorrect real classification. Conversely, an analysis of false negatives (rows 3-4) reveals a distinct bias against sensationalism; genuine articles employing clickbait triggers like "*SHOCKING*" are unjustly penalized, while legitimate scientific content regarding "*NASA*" is misclassified due to semantic overlaps in the training data.

Table 6. Examples of unanimous model errors

#	Type	Input Content (Summary)	True	Pred	Explanation (Based on Logs)
1	FP	"Treasury Department issues Directive 404 on Crypto..." (Fake Report)	Fake	Real	Totally deceived: Bureaucratic style ("Directive", "Treasury") completely fooled the model.
2	FP	"Archives reveal The Titanic was sunk by U-boat..." (Fake History)	Fake	Real	Deceived: The narrative style using "archival evidence" created a high sense of credibility.
3	FN	"SHOCKING: You won't believe what the President said!" (Clickbait)	Real	Fake	Unjustly blocked: The keyword "SHOCKING" caused the model to equate real news with spam/fake.
4	FN	"NASA's Perseverance rover collects new rock samples..." (Real Science)	Real	Fake	Unexpected error: Likely because keywords like "Mars/Ancient life" frequently appear in fake news datasets.

5.3. DISCUSSION

Our results demonstrate that SLMs are highly effective, often rivaling their larger counterparts.

- High-resource contexts: On the long-text WELFake dataset, distilled models proved exceptionally capable. Specifically, DistilBERT with full fine-tuning achieved an $F1$-score of 98.3%, which is statistically comparable to the teacher model RoBERTa-large 99.5%. This indicates that the semantic knowledge distilled into SLMs is sufficient for accurate detection in standard news articles.
- Low-resource/noisy contexts: On the short text dataset (LIAR), SLMs equipped with specialized adaptation strategies (adapters) achieved an $F1$-score of approximately 67.8%, significantly outperforming the BERT-base teacher 63.1%. The presence of substantial noise shifts the locus of generalization toward smaller, well-regularized models, away from over-parameterized LLMs.

The impact of PEFT varies critically by method and data characteristics:

- Bottleneck adapters (most robust): This strategy emerged as the most resilient. It was the only method capable of preventing model collapse on the challenging LIAR dataset, maintaining an $F1$–score of approximately 68%, significantly higher than that of 45.7% for LoRA, and performed consistently well on imbalanced data. Thus, the structural regularization of inserting new layers aids generalization.
- LoRA (efficiency under stability): In balanced experimental settings, such as WELFake, LoRA maintained competitive performance. However, its robustness degraded when exposed to short and noise-dominated inputs. This limitation becomes evident in the LIAR experiments, where the optimization process fails to converge reliably.
- Prompt tuning (sensitivity): Prompt tuning exhibits pronounced susceptibility to class imbalance, leading to substantial performance degradation under skewed label distributions. On FakeNewsNet, this results in an $F1$-score of approximately 65%, effectively restricting its utility to datasets with near-uniform class proportions.

We studied the trade-offs between inference time, model size, and computational cost. This led us to find the following three different profiles, each with its own set of traits and consequences:

- ALBERT is a good example of the storage vs. latency paradox. Downsizing the model to approximately 47MB (with 11 million parameters) does not always reduce the inference times. ALBERT is great for devices with limited ROM. This gives it a big advantage over relatively bigger models. However, due to its parameter-sharing architecture, its computational cost remains high at approximately 22.5 GFLOPs, resulting in a latency of 10.64 ms, slower than even the teacher models on long sequences.
- MiniLM (the efficiency sweet spot): This model demonstrates the optimal balance for real-time systems. By reducing computational cost to approximately 10.8 GFLOPs through self-attention optimization, it achieves a minimal latency of 0.31 ms on short text and processes over 500 samples/second on long text. It offers an approximately 2.9x higher speed than BERT while retaining near-teacher accuracy.
- Adapters (stability–throughput trade-off): Adapters preserve robust behavior on noise-heavy data such as LIAR, albeit at the expense of inference speed, which is constrained by the sequential execution of the added layers. For instance, MiniLM-adapter processes approximately 300 samples/second, approximately 60% slower than its full fine-tuning counterpart (approximately 860 samples/second), representing a strategic trade-off between maximum speed and detection reliability.

Finally, we synthesized our findings into a trade-off analysis to identify the optimal deployment configurations.

Fig. 3 demonstrates that most SLMs attain strong performance on WELFake ($F1 > 96\%$). However, MiniLM with full fine-tuning uniquely balances accuracy and speed, reaching inference rates of approximately 519 samples/s. This balance was not observed for ALBERT, whose shared-layer architecture leads to increased latency despite its compact parameterization.

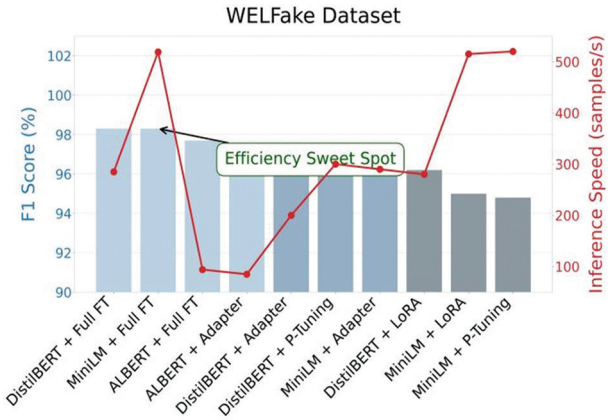


Fig. 3. Trade-off analysis on the WELFake Dataset, highlighting the efficiency sweet spot

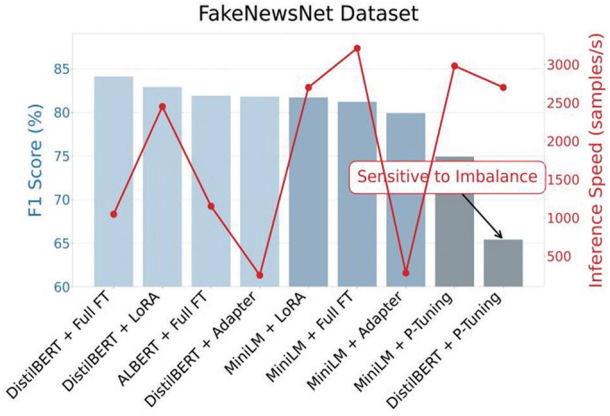


Fig. 4. Trade-off analysis on the FakeNewsNet Dataset, showing the impact of imbalance on prompt tuning

The results on the imbalanced FakeNewsNet dataset in Fig. 4 reveal a critical vulnerability in input-based optimization. Prompt tuning exhibits significant performance degradation compared to the weight-based methods (full fine-tuning and adapters). While MiniLM maintains the highest throughput, the trade-off in accuracy makes prompt tuning unsuitable for highly imbalanced scenarios.

On the challenging short-text LIAR dataset in Fig. 5, we observed a model collapse phenomenon where standard fine-tuning and LoRA fail to generalize. Bottleneck adapters proved to be uniquely robust, achieving the highest stability ($F1 \approx 68\%$) and preventing overfitting. Although adapters introduce a slight latency penalty compared to prompt tuning, they are the only viable strategy for effectively processing noisy, short-text data.

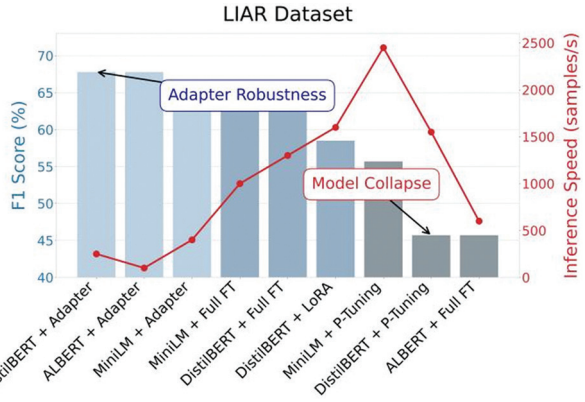


Fig. 5. Trade-off analysis on the LIAR Dataset, showing the robustness of adapters and the collapse of other models

6. CONCLUSION

In this work, we introduced SLM-FND, a lightweight yet effective framework based on fine-tuning SLMs for detecting fake news. Experiments conducted on benchmark datasets showed that the proposed method achieves competitive performance to larger models with significantly fewer trainable parameters, smaller inference times, and lower hardware cost. Furthermore, PEFT methods promote adaptability while significantly minimizing the computational cost. These results demonstrate the potential of high-performance SLMs to the problem of FND in edge and low-resource settings.

In the future, we plan to expand this work in the following directions: First, we will explore lightweight multimodal models to detect sophisticated fake news, including images and text. Second, we aim to explain this model using techniques like LIME and SHAP to enhance its understanding. Finally, detecting multilingual and multidisciplinary fake news is a potential research task.

DECLARATION OF THE USE OF AI TOOLS

During the revision process of this manuscript the authors used grammarly.com in order to check for spelling and grammar before submitting to Elsevier Language Editing services. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the content of the manuscript.

REFERENCES:

[1] World Health Organization, "Coronavirus Disease (COVID-19) Advice for the Public: Mythbusters", https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public/mythbusters (accessed: 2025)

[2] J. Zarocostas, "How to Fight an Infodemic", The Lancet, Vol. 395, No. 10225, 2020, p. 676. doi: 10.1016/S0140-6736(20)30461-X.

[3] S. Vosoughi, D. Roy, S. Aral, "The Spread of True and False News Online", Science, Vol. 359, No. 6380, 2018, pp. 1146-1151. doi: 10.1126/science.aap9559.

[4] R. Oshikawa, J. Qian, W. Y. Wang, "A Survey on Natural Language Processing for Fake News Detection", Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11-16 May 2020, pp. 6086-6093. doi: 10.48550/arXiv.1811.00770.

[5] J. Achiam et al. "GPT-4 Technical Report", arXiv:2303.08774, 2023.

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, USA, 2-7 June 2019, pp. 4171-4186. doi: 10.18653/v1/N19-1423.

[7] G. Hinton, O. Vinyals, J. Dean, "Distilling the Knowledge in a Neural Network", arXiv:1503.02531, 2015.

[8] V. Sanh et al. "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter", arXiv:1910.01108, 2019.

[9] W. Wang et al. "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers", Advances in Neural Information Processing Systems, Vol. 33, Virtual Conference (Vancouver, BC, Canada), 6-12 December 2020, pp. 5776-5788. doi: 10.48550/arXiv.2002.10957.

[10] Z. Lan et al. "ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations", Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26-30 April 2020. doi: 10.48550/arXiv.1909.11942.

[11] Y. Liu et al. "RoBERTa: A Robustly Optimized BERT Pre-training Approach", arXiv:1907.11692, 2019.

[12] S. Han, H. Mao, W. J. Dally, "Deep Compression: Com-pressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding", Proceedings of the 4th International Conference on Learning Representa-tions, San Juan, Puerto Rico, 2-4 May 2016. doi: 10.48550/arXiv.1510.00149.

[13] J. F. R. Bezerra, A. Kozierkiewicz, M. Pietranik, "A Novel Approach for Tweet Similarity in a Context-Aware Fake News Detection Model", IEEE Access, Vol. 13, 2025, pp. 57043-57061. doi: 10.1109/ACCESS.2025.3554540.

[14] N. Raza, S. J. Abdulkadir, Y. A. Abid, S. S. Albouq, A. Al-wadain, A. U. Rehman, E. H. Sumiea, M. Farhan, "Enhanc-ing Fake News Detection with Transformer-Based Deep Learning: A Multidisciplinary Approach", PLoS One, Vol. 20, No. 9, 2025, p. e0330954. doi: 10.1371/journal.pone.0330954.

[15] F. Kou, B. Wang, H. Li, C. Zhu, L. Shi, J. Zhang, L. Qi, "Po-tential Features Fusion Network for Multimodal Fake News Detection", ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 21, No. 3, 2025, pp. 1-24. doi: 10.1145/3711866.

[16] G. Salton, C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval", Information Processing & Management, Vol. 24, No. 5, 1988, pp. 513-523. doi: 10.1016/0306-4573(88)90021-0.

[17] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, "Fake News Detection on Social Media: A Data Mining Perspective", SIGKDD Explorations Newsletter, Vol. 19, No. 1, 2017, pp. 22-36. doi: 10.1145/3137597.3137600.

[18] H. Ahmed, I. Traore, S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques", Proceedings of the 1st International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, Vancouver, BC, Canada, 26-28 October 2017, Springer, 2017, pp. 127-138. doi: 10.1007/978-3-319-69155-8_9.

[19] Y. Wang et al. "EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection", Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19-23 August 2018, pp. 849-857. doi: 10.1145/3219819.3219903.

[20] S. Singhania, N. Fernandez, S. Rao, "3HAN: A Deep Neural Network for Fake News Detection", Proceedings of the 24th International Conference

on Neural Information Processing, Guangzhou, China, 14-18 November 2017, pp. 572-581. doi: 10.1007/978-3-319-70096-0_59.

[21] H. A. Asiri, M. Shoaib, "Arabic Fake News Detection on X (Twitter) Using Bi-LSTM Algorithm and BERT Embedding", IEEE Access, Vol. 13, 2025, pp. 189372-189386. doi: 10.1109/ACCESS.2025.3628530.

[22] S. Khalid, S. Ramzan, M. M. Iqbal, N. Thalji, A. Raza, A. Smerat, C. Kim, N. L. Fitriyani, M. Syafrudin, "Harnessing Interpretable Novel Combination of GloVe Embedding with Deep CNN-BiLSTM Neural Network for Fake News Detection", PLoS One, Vol. 20, No. 9, 2025, p. e0330154. doi: 10.1371/journal.pone.0330154.

[23] Y. Kim, "Convolutional Neural Networks for Sentence Classification", Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25-29 October 2014, pp. 1746-1751. doi: 10.3115/v1/D14-1181.

[24] N. Ruchansky, S. Seo, Y. Liu, "CSI: A Hybrid Deep Model for Fake News Detection", Proceedings of the 26th ACM International Conference on Information and Knowledge Management, Singapore, 6-10 November 2017, pp. 797-806. doi: 10.1145/3132847.3132877.

[25] C. Kumar, M. Bansal, M. A. Khan, V. Kaushik, M. Arquam, A. Alabdultif, "Graph-Augmented Transformer Ensemble Framework for Robust and Scalable Fake News Detection in Social Media Ecosystems", Scientific Reports, Vol. 16, 2025. doi: 10.1038/s41598-025-31653-3.

[26] K. S. F. Ahmad, S. G. Pamidimukkala, A. P. Sathe, S. GNVG, K. Ch et al. "Hybrid Optimization Driven Fake News De-tection Using Reinforced Transformer Models", Scientific Reports, Vol. 15, No. 1, 2025, pp. 1-16. doi: 10.1038/s41598-025-99936-3.

[27] R. K. Kaliyar, A. Goswami, P. Narang, "FakeBERT: Fake News Detection in Social Media with a BERT-Based Deep Learning Approach", Multimedia Tools and Applications, Vol. 80, No. 8, 2021, pp. 11765-11788. doi: 10.1007/s11042-020-10183-2.

[28] B. Hu, Q. Sheng, J. Cao, Y. Shi, Y. Li, D. Wang, P. Qi, "Bad Actor, Good Advisor: Exploring the Role of Large Lan-guage Models in Fake News Detection", Proceedings of the 38th AAAI Conference on Artificial Intelligence, Van-couver, BC, Canada, 20-27 February 2024, Vol. 38, 2024, pp. 22105-22113. doi: 10.1609/aaai.v38i20.30214.

[29] Z. Tong, Y. Gu, H. Liu, Q. Liu, S. Wu, H. Shi, X.-Y. Zhang, "Generate First, Then Sample: Enhancing Fake News De-tection with LLM-Augmented Reinforced Sampling", Proceedings of the 63rd Annual Meeting of the Associa-tion for Computational Linguistics, Vienna, Austria, 27 Ju-ly - 1 August 2025, Vol. 1, 2025, pp. 24276-24290. doi: 10.18653/v1/2025.acl-long.1182.

[30] A. Sittar, M. Smiljanic, A. Gucek, M. Grobelnik, "Fake News Detection Through LLM-Driven Text Augmenta-tion Across Media and Languages", Machine Learning and Knowledge Extraction, Vol. 8, No. 4, 2026. doi: 10.3390/make8040103.

[31] X. Yang, Y. Wang, J. Zhu, P. Ding, H. Liu, X. Zhang, H. Liu, "Cross-Domain Fake News Detection on Unseen Domains via LLM-Based Domain-Aware User Modeling", arXiv:2602.01726, 2026.

[32] J. Wang, Z. Zhu, C. Liu, R. Li, X. Wu, "LLM-Enhanced Multimodal Detection of Fake News", PLoS One, Vol. 19, No. 10, 2024, p. e0312240. doi: 10.1371/journal.pone.0312240.

[33] M. A. Habib, M. A. H. Wadud, M. F. Mridha, M. J. Hossen, "LLM-Powered Multimodal Reasoning for Fake News Detection", Computers, Materials, & Continua, Vol. 87, No. 1, 2026. doi: 10.32604/cmc.2025.070235.

[34] R. Bharati, J. Bharti, V. Dehalwar, "A Novel Fuzzy Logic-Based Hybrid Framework for Detecting Fake News", Discover Computing, Vol. 29, No. 1, 2026, p. 58. doi: 10.1007/s10791-025-09882-x.

[35] S. Tahmasebi, E. Müller-Budack, R. Ewerth, "Robust Fake News Detection Using Large Language Models Under Adversarial Sentiment Attacks", arXiv:2601.15277, 2026.

[36] E. J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models", Proceedings of the 10th International Conference on Learning Representations, Virtual Event, 25-29 April 2022. doi: 10.48550/arXiv.2106.09685.

[37] N. Houlsby et al. "Parameter-Efficient Transfer Learning for NLP", Proceedings of the 36th Inter-

national Conference on Machine Learning, Long Beach, CA, USA, 9-15 June 2019, pp. 2790-2799. doi: 10.48550/arXiv.1902.00751.

[38] B. Lester et al. "The Power of Scale for Parameter-Efficient Prompt Tuning", Proceedings of the Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7-11 November 2021, pp. 3045-3059. doi: 10.18653/v1/2021.emnlp-main.243.

[39] P. K. Verma et al. "WELFake: Word Embedding over Linguistic Features for Fake News Detection", IEEE Transactions on Computational Social Systems, Vol. 8, No. 4, 2021, pp. 881-893. doi: 10.1109/TCSS.2021.3068519.

[40] K. Shu et al. "FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media", Big Data, Vol. 8, No. 3, 2020, pp. 171-188. doi: 10.1089/big.2020.0062.

[41] W. Y. Wang, "'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection", Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July - 4 August 2017, Vol. 2, pp. 422-426. doi: 10.18653/v1/P17-2067.

[42] T. Wolf et al. "Transformers: State-of-the-Art Natural Language Processing", Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16-20 November 2020, pp. 38-45. doi: 10.18653/v1/2020.emnlp-demos.6.

[43] J. Dongarra, J. Gunnels, H. Bayraktar, A. Haidar, D. Ernst, "Hardware Trends Impacting Floating-Point Computations in Scientific Applications", arXiv:2411.12090, 2024.

[44] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, Vol. 12, 2011, pp. 2825-2830. doi: 10.5555/1953048.2078195.