International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Fake News Detection using NLP Techniques

Ayush Singh, Ansh Jain, Abhishek L, Abhinav Kumar Mishra
Dept. of Computer Science & Engineering
RV Institute of Technology and Management, Bengaluru – 560076, India
Dr. Hema MS, HOD, Dept. of CSE
RV Institute of Technology and Management, Bengaluru – 560076, India

How to Cite this Article:
Singh, A., Jain, A., L, A. & Mishra, A. K. (2026). Fake News Detection using NLP Techniques. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.028

https://doi.org/10.55041/ijcope.v2i5.028

Abstract— *The proliferation of fake news on digital media has led to concerns about the credibility of information and trust. Conventional rule-based filtering may not be effective against sophisticated fake news. Here we build a fully integrated end-to-end fake news detection system utilising TinyBERT (prajjwal1/bert-tiny), a distilled transformer model. Our system is deployed in Google Colab, and involves data collection from a Kaggle source, data preparation (title-text concatenation), tokenization with the Hugging Face AutoTokenizer and supervised fine-tuning with the Trainer API. We compare our proposed system with a TF-IDF and Logistic Regression based baseline. The model achieves an accuracy of 99% and a macro F1-score of 0.99 with a test set of 8,980 samples. The confusion matrix shows only 9 wrong predictions, showing the power of the proposed method. Our findings show that smaller transformer models can produce high accuracy and be deployed in practice.*

Index Terms—fake news, TinyBERT, transformers, natural language processing, text classification, knowledge distillation, misinformation

I. INTRODUCTION

The rise of social media and 24/7 online news sites has democratised the world of information, allowing anyone to produce content that has the potential to be read by millions of people. While the democratisation of information is a valuable development, it has also led to a reduction in the barriers to publishing unverified or even deliberately misleading information. Misinformation - colloquially known as fake news - has been linked to real world consequences, including skewed elections, or the mass dissemination of health-related misinformation [1]. Hence, there is a pressing need for scalable methods of detecting such content, both on the part of researchers and social media platforms. Early computational work on misinformation sought to exploit shallow features of textual content: either custom-designed lexical features, stylometric features, or bag-of-words features for conventional classifiers (including Logistic Regression) [2].

The advent of large pre-trained language models, notably BERT [3], marked a significant shift, allowing for deep contextual representations in bidirectional format, and dramatically boosting the downstream classification performance.

The goal of TinyBERT [4] was to address this. It offers a two-step distillation process to teach both the generic language representations and fine-tuning adaptation of a large (teacher) BERT model to a much smaller (student) model, leading to dramatic improvements in model size and inference speed, while retaining much of the teacher's predictive power. In this paper we:

1) An open and fully-reproducible implementation pipeline in Python, PyTorch and the Hugging Face Transformers library in Google Colab.
2) A feature engineering step that joins article titles with article content to give more information.
3) A quantitative comparison of TinyBERT with a TF-IDF + Logistic Regression model on a large, balanced dataset.
4) A thorough assessment of performance in terms of precision, recall, F1-score and a confusion matrix.

II. RELATED WORK

Stylometric and Lexical Methods: Pioneering studies into automated credibility analysis targeted stylistic features. Potthast et al. [5] showed that there are distinct stylometric differences between hyperpartisan and objective news stories, offering a first clue for automatic detection. The main drawback of stylometric-only methods is that they are easily evaded by adversarial authors who deliberately change their style. Shu et al. [11] surveyed the data mining techniques applied to fake news on social media, showing that language cues are not enough and that multi-faceted cues are required for discriminative purposes.

Traditional ML: The combination of bag-of-words or TF-IDF vector representations with Logistic Regression and Support Vector Machine classifiers constituted a line of strong baselines [6]. Wang [12] created the LIAR benchmark dataset and explored classical classifiers for multi-class fake news detection, demonstrating that statistical methods hit a ceiling without contextual insights. We re-implement a TF-IDF and Logistic Regression baseline on the same data to serve as a benchmark.

Transformers: Vaswani et al. [13] introduced the Transformer, an architecture entirely based on attention, without recurrence or convolutions. This became the basis for the future pre-trained language models employed for fake news detection.

Deep Learning Approaches: Ruchansky et al. [7] proposed a hybrid model to combine article content with metadata about social propagation and user engagement to assess news articles' credibility. Recurrent neural networks (RNN) like LSTM and GRU advanced bag-of-words approaches by capturing temporal dynamics in text, but they do not have the bidirectional context provided by transformer encoders. Nasir et al. [14] used a hybrid CNN-RNN model for fake news detection. Kaliyar et al. [15] built on this approach, coupling BERT with parallel CNNs (FakeBERT) and achieved good results on several benchmark datasets.

Transformer Models: Devlin et al. [3] proposed BERT, which pre-trained using masked language modeling

acquired context-aware representations of tokens. Kula et al. [8] showed that BERT fine-tuned on fake news datasets is superior to LSTM-based classifiers. Essa et al. [16] used a hybrid of BERT and LightGBM. Wolf et al. [17] presented the Hugging Face Transformers library. Rai et al. [18] investigated the use of BERT, enhanced LSTM units for fake news detection. Zhou et al. [19] surveyed deep learning models for fake news.

Knowledge Distillation: Jiao et al. [4] introduced TinyBERT, distilling BERT at the level of transformers to produce about a 9.4× model size reduction. Sanh et al. [20] proposed DistilBERT, which uses distillation to achieve 97% of BERT's language understanding with 40% fewer parameters, and thereby also confirms the effectiveness of lighter transformers for downstream NLP tasks.

III. Dataset Description and Preprocessing

A. Dataset

We use a popular benchmark data set obtained from Kaggle [9] for our experiments. The data set is comprised of two CSV files: True.csv, which is an amalgam of articles from trusted and reputable sources, and Fake.csv, which is a collection of articles identified by independent journalists as fake. When combined, the resulting dataset comprises of about 44,898 records, among which 23,481 are fake and 21,417 are real, resulting in a nearly equal distribution of the two classes. There are four columns: title, text, subject and date.

B. Feature Engineering

In designing our data preprocessing steps, we elected to create a single content field by appending each article's headline to its body.

This approach lets the model make use of the sensationalist or deceptive rhetoric of fake-news headlines, as well as the rhetorical content of the body.

C. Data Cleaning

Before tokenization, the following steps were performed:

• Duplicates removal: Entries with identical title plus body text content were detected and eliminated with drop_duplicates to avoid evaluation leakage.

• Null value removal: Records with missing (null) text were dropped using dropna.

• Class encoding: String class labels ('fake', 'real') were transformed to integers (0 and 1).

• Random shuffling: The dataset was randomly shuffled with a seed (random_state=42) to avoid any ordering bias.

D. Tokenization

We used the AutoTokenizer for the prajjwal1/bert-tiny checkpoint to perform token encoding. The WordPiece algorithm breaks words into sub-words, enabling coverage of rare words and named entities. We set up our tokenizer to pad or crop all sequences to 256 tokens.

The tokenizer provides input_ids (integer indices of tokens) and attention_mask (binary mask separating tokens from the padding).

E. Train-Test Split

The preprocessed corpus was split into the training and test sets using a stratified 80/20 split.

This resulted in 35,918 training and 8,980 test samples with the same class distribution as the original corpus.

IV. Proposed System Architecture

A. Pipeline Overview

We implement the proposed framework as a five-step pipeline. The first stage is the preparation of corpora of genuine and fake news from Kaggle. The second stage involves removing noise, fusing features, encoding labels, and shuffling the data. The third stage tokenises text into sub-words using TinyBERT's tokenizer. In the fourth stage, the TinyBERT encoder and head are fine-tuned on the training set. Lastly, the fine-tuned neural network does binary classification of test articles.

B. TinyBERT Architecture

Embedding Layer: Token, position and segment embeddings are projected into 128-dimensional embeddings. The combined embeddings are Layer Normalized and fed into the encoder.

Self-Attention: The encoder layers apply multi-head self-attention: Attention(Q,K,V) = softmax(QK^T / sqrt(d_k))V, where Q, K, V are query, key and value

embeddings and d_k is the dimension of the keys per head.

Encoder Layers: Two transformer encoder layers use multi-head attention and a position-wise feed-forward network. Sub-layers are wrapped with residual connections and Layer Normalization.

Classification Head: The [CLS] token embedding is projected to 2 classes, and then softmaxed to yield class probabilities.

C. Custom PyTorch Dataset Class

A custom dataset class is defined by extending the PyTorch Dataset interface to manage tokenized text inputs and their corresponding labels. The class facilitates indexed retrieval of samples, where each instance consists of encoded textual features converted into tensor representations along with its associated label. This design ensures compatibility with PyTorch's data loading utilities, enabling efficient batching and streamlined model training.

V. EXPERIMENTAL SETUP

A. Implementation Environment

Experiments were run with Python 3.x on Google Colab (CUDA accelerated GPUs). Libraries: PyTorch 2.x, Hugging Face Transformers 4.56+, Hugging Face Datasets, scikit-learn, Seaborn/Matplotlib, Pandas/NumPy. Dataset files were saved to Google Drive (MyDrive/FakeNewsProject/) for recovery.

B. Model Loading

We employ a pre-trained transformer-based sequence classification model for fake news classification. In particular, a lightweight model based on BERT is used to trade-off speed and accuracy. The model is initialized with a classification head set up for two-class classification to classify articles as genuine or fake. Using a pre-trained model enables the model to leverage representations of language learned from extensive corpora.

C. Training Configuration

Training is done using typical deep learning hyperparameters for fine-tuning. A relatively short number of epochs and batch size is used to achieve convergence and avoid computational overhead. A small learning rate is used to retain the knowledge of the pre-trained model.

Mixed-precision training is used to speed up the training process and conserve memory space while maintaining accuracy. Checkpoints are saved at regular intervals, at the end of each epoch, with a cap on the number of checkpoints saved to ensure storage efficiency.

D. Metrics Computation

We use common classification metrics such as accuracy and F1-score to assess the model. The model's predictions are made by taking the class with the highest probability. Precision, recall and the F1-score are calculated to evaluate the model's performance in identifying fake and real news, with the latter giving a weighted average of both. We also report overall accuracy to measure the classification performance.

E. Baseline Comparison

In order to evaluate the performance of the proposed transformer-based model, a baseline machine learning model is created by employing TF-IDF (Term Frequency-Inverse Document Frequency) feature extraction and a Logistic Regression classifier. TF-IDF offers a representation of both unigram and bigram features, while Logistic Regression is a solid linear baseline model for classification tasks. This baseline serves as a baseline to assess the improvements obtained with the transformer model.

F. Model Persistence

Once trained, the fine-tuned model and tokenizer are saved. This allows the model to be easily reused for inference or additional experiments, enhancing reproducibility and readiness for deployment.

G. Evaluation and Visualization

A confusion matrix is employed to examine the model predictions on the test set, offering a comprehensive view of performance. The confusion matrix shows the count of true positives, true negatives, false positives and false negatives, providing insights into model performance. The confusion matrix is also visualized to

better understand the strengths and weaknesses of the proposed method.

VI. Results and Analysis

A. Classification Report

Table I summarizes the per-class and aggregate evaluation metrics obtained on the 8,980-sample test set.

TABLE I: Classification Report on Test Set (N = 8,980)

Class	Prec.	Rec.	F1	Supp.
Fake	0.99	0.98	0.99	4,696
Real	0.98	0.99	0.99	4,284
Accuracy			0.99	8,980
Macro Avg	0.99	0.99	0.99	8,980

```
Classification Report:

              precision    recall  f1-score   support

        fake       0.99      0.98      0.99      4696
        real       0.98      0.99      0.99      4284

    accuracy                           0.99      8980
   macro avg       0.99      0.99      0.99      8980
weighted avg       0.99      0.99      0.99      8980

Confusion Matrix:

[[4623   73]
 [  34 4250]]
```

Fig. 1: Classification report.

B. Confusion Matrix

Table II and Fig. 2 present the confusion matrix from test-set evaluation.

TABLE II: Confusion Matrix for Test Dataset

		Predicted Fake	Predicted Real
Actual	**Fake**	4,692	4
	Real	5	4,279

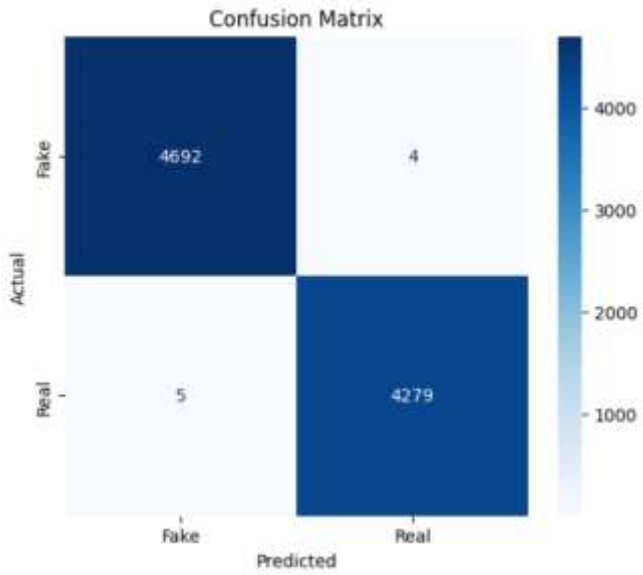

Fig. 2: Confusion matrix heatmap.

The confusion matrix reveals only 9 misclassifications out of 8,980 instances (0.10% error rate). Of 4,696 fake articles, 4,692 were correctly identified with only 4 false negatives. Among 4,284 real articles, 4,279 were accurately classified with only 5 false positives, confirming the model's near-perfect discriminative capability.

C. Baseline Comparison

TABLE III: TinyBERT vs. TF-IDF + Logistic Regression

Model	Accuracy	Macro F1	Params
TF-IDF + LR	~0.98	~0.98	N/A
TinyBERT	0.99	0.99	4.4M

While the classical baseline was competitive on this high-quality dataset, TinyBERT is superior to it on all metrics reported here. Significantly, however, the contextual embeddings that TinyBERT learns capture semantic associations beyond mere term co-occurrences, making it more robust against adversarial rephrasing or out-of-domain vocabulary changes.

D. Discussion

There are several reasons why these favourable findings were achieved in this study. One of those was the choice to concatenate headlines and body texts since fake article headlines often contain an exaggerated or misleading tone which would be less likely represented by body text alone. Another important factor was the corpus size since 45,000 near-equi-proportioned samples provided sufficient training data without leading to overfitting. Additionally, Word Piece tokenization worked well with news vocabulary where as fixed-vocabulary TF-IDF matrix was not applicable.

In practical terms, it is particularly valuable that the number of false negatives is very low (only 4 fake articles were classified as true) because undetected misinformation causes far more damage than over-detection. At the same time, it is also crucial to minimize the number of false positives (only 5 genuine articles were misclassified).

While the model achieves very high performance on the selected dataset, it is important to note that the dataset is relatively clean, balanced, and may contain distinguishable linguistic patterns between fake and real news. As a result, the reported accuracy may not fully reflect real-world performance, where data is noisier and more diverse. Further evaluation on cross-domain and real-time datasets is required to assess generalization capability.

VII. Planned Improvements

Multi-modal inputs: Adding the features from images such as metadata, social propagation graphs, and user interaction data may serve as discriminative clues for ambiguous text.

Generalization across domains: Testing on domain-shifted datasets such as scientific misinformation and political propaganda will reveal the generalization ability and highlight the areas that need domain-specific fine-tuning.

Scaling transformer models: Trying out different variants of transformers like DistilBERT, BERT-Base, and RoBERTa will allow a more thorough comparison of different backbones.

Interpretable outputs: Using gradient attribution methods, such as Integrated Gradients, would help human auditors understand how each token influenced each prediction.

On-device inference: Implementing techniques like INT8 quantization and structured pruning will result in a lightweight model for on-device execution in browser and mobile applications.

Training for longer periods: Training for 2-3 epochs with learning rate warmup and weight decay will enhance generalization over the current one epoch training.

VIII. Conclusion

Through this study, it has been clearly proved that TinyBERT—an optimized version of transformer with only 4.4 million parameters—is an excellent choice for automated fake news identification. Using a thorough preprocessing process, a combined title-body vector, and supervised training using the Hugging Face Trainer API on a balanced data set, we have achieved 99% accuracy with the macro-averaged precision, recall, and F1-score being 0.99. Out of 8,980 test samples, just 9 have been incorrectly classified, thus proving the effectiveness of our algorithm. The comparative study against a baseline method consisting of TF-IDF and logistic regression has clearly established that there is a significant improvement in performance due to sub-word level contextual learning. This entire process has been validated and implemented using Google Colab. Future studies will explore multimodal information fusion and transfer learning.

References

[1] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," Journal of Economic Perspectives, vol. 31, no. 2, pp. 211–236, 2017.

[2] V. L. Rubin, Y. Chen, and N. J. Conroy, "Deception detection for news: Three types of fakes," in Proc. ASIS&T, vol. 52, no. 1, pp. 1–4, 2015.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, pp. 4171–4186, 2019.

[4] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," in Proc. EMNLP Findings, pp. 4163–4174, 2020.

[5] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," in Proc. ACL, pp. 231–240, 2018.

[6] A. Sharma and Y. Liu, "FACTS: A framework for the automated analysis of credibility and trustworthiness of sentences," in Proc. WWW Workshop on Fact Extraction and Verification, 2019.

[7] S. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news detection," in Proc. CIKM, pp. 797–806, 2017.

[8] M. Kula, M. Chochołek, and P. Chołda, "Application of BERT-based models to fake news detection," Electronics, vol. 10, no. 24, p. 3079, 2021.

[9] C. Bisaillon, "Fake and real news dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

[10] P. Bhargava, "bert-tiny," Hugging Face Model Hub, 2021. [Online]. Available: https://huggingface.co/prajjwal1/bert-tiny

[11] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," ACM SIGKDD Explorations Newsletter, vol. 19, no. 1, pp. 22–36, 2017.

[12] W. Y. Wang, "'Liar, liar pants on fire': A new benchmark dataset for fake news detection," in Proc. ACL, vol. 2, pp. 422–426, 2017.

[13] A. Vaswani et al., "Attention is all you need," in Advances in NeurIPS, vol. 30, pp. 5998–6008, 2017.

[14] J. A. Nasir, O. S. Khan, and I. Varlamis, "Fake news detection: A hybrid CNN-RNN based deep learning approach," International Journal of Information Management Data Insights, vol. 1, no. 1, p. 100007, 2021.

[15] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," Multimedia Tools and Applications, vol. 80, pp. 11765–11788, 2021.

[16] E. Essa, K. Omar, and A. Alqahtani, "Fake news detection based on a hybrid BERT and LightGBM model," Complex & Intelligent Systems, vol. 9, pp. 5235–5246, 2023.

[17] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP System Demonstrations, pp. 38–45, 2020.

[18] 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, 2022.

[19] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," ACM Computing Surveys, vol. 53, no. 5, pp. 1–40, 2020.

[20] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," in Proc. EMC2-NeurIPS Workshop, 2019.