NMIRJ

# Nrityanjali Multidisciplinary International Research Journal

Volume 1, Issue 1, October 2025

**E-ISSN:** Applied **Website:** https://nrityanjaliresearchjournal.com/home/ **Email:** info@nrityanjaliresearchjournal.com

## "A Comparative Study of Machine Learning Models for Fake News Detection Using Python and NLP"

**Mayank Ratmele**

*Information Technology (AIR) Department,*

*Madhav Institute Of Technology & Science,Gwalior*

**Abstract**

The rampant spread of fake news in digital media poses a serious threat to public trust, democratic stability, and societal well-being. This study focuses on developing an automated fake news detection system using Python and Natural Language Processing (NLP) techniques. Employing benchmark datasets such as the Fake and Real News Dataset and the LIAR dataset, the research explores both traditional machine learning classifiers (Logistic Regression, Naïve Bayes, SVM, Random Forest, XGBoost) and advanced deep learning models (LSTM, BiLSTM, and BERT). Text preprocessing techniques like tokenization, stopword removal, lemmatization, and vectorization (TF-IDF, Word2Vec, BERT embeddings) are applied to convert raw news text into machine-readable features. Performance evaluation using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC indicates that transformer-based BERT significantly outperforms other models, achieving an accuracy of 96.2% and an AUC of 0.973. The findings highlight the potential of contextual embeddings in fake news detection and reinforce the effectiveness of Python's NLP ecosystem in automating content credibility assessment. This research contributes to combating misinformation by offering scalable and accurate solutions for real-world deployment in digital platforms.

**Keywords:** Fake News Detection, Natural Language Processing, Python, Machine Learning, Deep Learning, BERT, TF-IDF, LSTM, Text Classification, Misinformation

### 1. Introduction

In today's digitized world, information travels faster than ever, often without sufficient verification or validation. With the advent of the internet and the widespread use of social media platforms, the dissemination of news and opinions has become more democratized—but also more vulnerable to manipulation. One of the most concerning manifestations of this phenomenon is the rise of fake news, which refers to deliberately fabricated information that mimics the appearance of legitimate news content with the intent to mislead or deceive (Zhou & Zafarani, 2020). The rapid spread of fake news can result in social unrest, political manipulation, and public health crises, as evidenced during major events like elections or the COVID-19 pandemic (Alam et al., 2021).

The fight against fake news has become an interdisciplinary challenge that spans journalism, psychology, linguistics, and computer science. Traditional journalistic methods of fact-checking and source verification, though vital, are no longer sufficient on their own due to the sheer volume of content being generated daily. Therefore, researchers and practitioners are increasingly turning to Natural Language Processing (NLP) and machine learning (ML) tools—particularly in Python—to automate the detection and classification of misleading or false information (Ahmed et al., 2017; Singh & Sharma, 2020).

NLP enables computers to understand, interpret, and manipulate human language. When combined with ML algorithms, it becomes possible to train models that can detect textual patterns and linguistic features commonly associated with fake news. For instance, fake news articles often contain exaggerated phrases, biased sentiment, and shallow lexical structures compared to legitimate news sources (Pérez-Rosas et al., 2018). Python, as a programming language, provides a rich ecosystem of libraries and frameworks—such as NLTK, SpaCy, Scikit-learn, TensorFlow, and Transformers—that make it ideal for implementing and testing NLP-based fake news detection systems (Singh & Sharma, 2020).

The importance of fake news detection lies in its implications for democratic societies. Inaccurate or intentionally misleading content can influence public opinion and policymaking. A notable example is the U.S. 2016 Presidential Election, where numerous reports indicated that fake news stories on Facebook reached more users than credible news during critical periods (Shu et al., 2017). The economic impact is also significant, as businesses and individuals often make decisions based on misleading financial or health-related information. Consequently, a scalable, automated, and accurate method to detect fake news is not just a technical problem—it is a societal necessity (Kumar & Shah, 2018).

Over the years, various datasets have been developed to support research in this domain. One of the most widely used datasets is the LIAR dataset, compiled by Wang (2017), which includes thousands of labeled short statements from political figures. The dataset categorizes content into six classes—ranging from "pants on fire" to "true"—and has been instrumental in evaluating model performance across fine-grained levels of deception. Similarly, the FakeNewsNet dataset developed by Shu et al. (2020) combines textual content with social engagement metadata such as likes, shares, and user comments. These datasets are often preprocessed in Python using techniques like tokenization, stopword removal, stemming, and vectorization (TF-IDF or Word2Vec) before being fed into classifiers (Ahmed et al., 2017).

Machine learning algorithms such as Naïve Bayes, Logistic Regression, Random Forests, and Support Vector Machines (SVMs) have been commonly used in the early stages of fake news detection. These models rely on statistical features such as term frequency and document length to distinguish fake from real articles (Ahmed et al., 2017). More recently, deep learning techniques like Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Convolutional Neural Networks (CNNs) have been deployed to capture complex sentence structures and contextual dependencies in news content (Kaliyar et al., 2021; Ruchansky et al., 2017). These models can automatically learn discriminative features from raw text, improving the generalization capability of fake news detectors.

The emergence of transformer-based models, particularly BERT (Bidirectional Encoder Representations from Transformers), has pushed the boundaries of NLP tasks, including fake news classification. Devlin et al. (2019) introduced BERT as a deep bidirectional model that learns contextual relations between words by jointly conditioning on both left and right contexts. This feature makes BERT highly effective in understanding the semantics of news articles and identifying deceptive cues buried within complex narratives. Fine-tuning BERT for fake news detection using Python libraries like Hugging Face's Transformers has yielded state-of-the-art performance in recent benchmarks (Zhou & Zafarani, 2020).

However, detecting fake news is not just a question of linguistic analysis. Contextual features such as the credibility of the news source, user engagement patterns, and temporal diffusion dynamics also play a vital role. Ruchansky et al. (2017) proposed the CSI model, a hybrid framework that incorporates content-based features, social context, and user behavior. This multifaceted approach reflects the complexity of real-world fake news detection, where a single piece of text must be evaluated not only for its content but also for its origin and impact.

An additional concern in fake news research is the multilingual and cross-domain adaptability of models. Most datasets and models are trained in English and may not generalize well across languages or topics. Pérez-Rosas et al. (2018) and Thorne & Vlachos (2018) highlighted the limitations of current datasets and called for more robust, diverse, and scalable solutions. The ethical implications of automated content labeling—such as censorship, algorithmic bias, and transparency—also demand attention (Zhang & Ghorbani, 2020).

In response to these challenges, researchers are now exploring hybrid models that combine content-based NLP techniques with metadata-based and network-based methods (Shu et al., 2017). Some recent studies have also focused on real-time detection capabilities to flag misleading content during breaking news events, particularly during crises like pandemics or natural disasters (Alam et al., 2021). Integrating sentiment analysis, topic modeling, and stance detection into fake news classifiers is another emerging direction (Pérez-Rosas et al., 2018).

To summarize, the current body of literature reveals a rapidly evolving field with promising advancements in text analysis, feature engineering, and neural language models. Python has emerged as the preferred platform due to its extensive support for NLP and machine learning. Nevertheless, significant gaps remain in areas such as model interpretability, low-resource language adaptation, and adversarial robustness. This research aims to contribute to this evolving landscape by building and evaluating Python-based NLP models for fake news detection, utilizing a range of traditional and deep learning approaches, benchmark datasets, and real-world validation scenarios.

## 2. Literature Review

The rise of digital media has intensified the challenge of distinguishing factual content from fabricated information. As fake news becomes more pervasive and impactful, especially through social media, research has increasingly focused on using Natural Language Processing (NLP) and machine learning to combat misinformation. This literature review explores foundational and contemporary contributions to fake news detection, analyzing methodologies, models, and datasets that leverage Python and NLP frameworks.

Ahmed et al. (2017) conducted one of the earlier studies applying n-gram textual analysis with machine learning to detect fake news, demonstrating that lexical features such as bigrams and trigrams could effectively distinguish fake from real news. Their work laid a structural foundation for using NLP-driven text representation combined with classifiers like Support Vector Machines (SVM) and Naïve Bayes in Python-based environments. In the same year, Shu et al. (2017) introduced a broader data mining perspective on fake news detection by integrating content features with user behavior on social media. Their proposed FakeNews Challenge (FNC) highlighted the importance of combining textual analysis with contextual metadata, setting the stage for future hybrid models.

Wang (2017) contributed significantly with the LIAR dataset, a benchmark comprising short statements labeled across six truthfulness categories. This dataset remains crucial in evaluating model robustness across fine-grained classes. Ruchansky et al. (2017) proposed the CSI model—an ensemble of content, social context, and user response features—demonstrating that integrating multi-modal inputs in deep learning frameworks outperforms traditional classifiers. Their model incorporated Long Short-Term Memory (LSTM) networks to understand temporal dependencies in news dissemination patterns.

Singh and Sharma (2020) conducted a Python-based implementation study that combined tokenization, stemming, and Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to preprocess datasets, followed by classification using Random Forest and Gradient Boosting. Their research showed high detection accuracy and reinforced Python's ecosystem—particularly with Scikit-learn and NLTK—as a reliable framework for such applications. Kumar and Shah (2018) extended the discourse by surveying false information phenomena on web platforms, suggesting that the problem of fake news is not merely technical but also sociological, requiring user-centric model training and domain-specific knowledge.

Kaliyar et al. (2021) advanced detection accuracy using a deep convolutional neural network (CNN)-based architecture called FNDNet. Their model outperformed traditional classifiers by automatically learning hierarchical patterns in news headlines and bodies. This research demonstrated that fake news often lacks linguistic depth and coherence, which deep learning models can detect more effectively. Meanwhile, Zhang and Ghorbani (2020) provided a comprehensive overview of online misinformation, emphasizing the evolving nature of fake news and advocating for adaptive models that account for temporal dynamics and platform-specific behaviors.

Alam et al. (2021) explored fake news propagation during the COVID-19 pandemic, proposing models to understand the psychological and thematic tendencies of fake news spreaders. Their work highlighted the role of sentiment and emotion detection using NLP in Python, where classifiers were trained on linguistic markers like exaggerated claims, fear-based language, and certainty expressions. Pérez-Rosas et al. (2018) introduced a dataset of news articles labeled by professional fact-checkers and built a logistic regression-based classifier using syntactic and stylistic features. Their results confirmed that fake news tends to use simpler language and more subjective tones.

Thorne and Vlachos (2018) addressed the task of automated fact-checking, categorizing it as a distinct but related challenge. They emphasized the complexity of verifying claims in real-time and proposed decomposing articles into check-worthy units. Their methodology included sentence-level fact extraction and cross-referencing with verified databases—techniques that rely heavily on Python-based NLP pipelines and named entity recognition (NER). Shu et al. (2020) developed FakeNewsNet, a data repository that aggregates news content, social context, and user interaction patterns. This resource enables researchers to explore the influence of social network structures on fake news diffusion.

Zhou and Zafarani (2020) provided one of the most comprehensive surveys on fake news detection, reviewing both content-based and context-based approaches. They discussed supervised and unsupervised learning models, emphasizing that deep learning architectures like BERT (Bidirectional Encoder Representations from Transformers) are setting new benchmarks. Conroy et al. (2015) were among the early advocates of linguistic and rhetorical feature analysis, suggesting deception cues such as hedging, passive voice, and lack of source attribution as indicators of fake news.

Devlin et al. (2019), the creators of BERT, revolutionized the NLP domain with a model capable of understanding deep semantic relationships between words and sentences. BERT's pre-trained language models can be fine-tuned for fake news classification, enabling higher contextual awareness than TF-IDF or n-gram-based methods. Its implementation in Python using Hugging Face's Transformers library has become a standard in state-of-the-art NLP projects, allowing researchers to classify news with significantly improved precision and recall.

Cumulatively, these studies underscore that fake news detection is a multifaceted problem requiring both linguistic insight and computational robustness. Traditional machine learning models using shallow features like TF-IDF, lexical markers, and metadata have yielded promising results. However, the current trend is shifting towards deep learning, especially transformer-based architectures that capture semantic nuances. Python remains the dominant programming language due to its mature ecosystem, open-source libraries (NLTK, SpaCy, TensorFlow, Scikit-learn, Hugging Face), and community support.

Furthermore, the reviewed literature emphasizes that dataset quality, annotation consistency, and domain relevance significantly influence model performance. Studies also highlight the importance of hybrid approaches—integrating user behavior, temporal patterns, and cross-platform data—to counter increasingly sophisticated fake news generation techniques like GPT-based text generators.

In conclusion, fake news detection using Python and NLP has evolved from rule-based and statistical methods to deep learning and transformer-based approaches. The reviewed literature not only validates the potential of automated detection systems but also reveals gaps in interpretability, cross-lingual adaptation, and real-time deployment. Future research should focus on explainable AI, model fairness, adversarial robustness, and ethical data use to ensure scalable and trustworthy solutions in the battle against misinformation.

## 3. Methodology

### 3.1 Research Design

This study employs an experimental and quantitative research design focused on the development and evaluation of machine learning models for the detection of fake news using Natural Language Processing (NLP) techniques in Python. The study involves data preprocessing, feature extraction, model training, testing, and performance evaluation using established classification metrics. The research is conducted in a structured pipeline that includes data collection, cleaning, NLP transformation, model implementation, and comparative analysis.

### 3.2 Dataset Selection

Two publicly available datasets are used in this research:

1. LIAR Dataset (Wang, 2017): Contains 12,836 short statements from political figures, labeled with truthfulness levels ranging from "pants on fire" to "true".
2. Fake and Real News Dataset (Kaggle): Includes 44,000 articles with binary labels (fake or real) and corresponding article text, title, and subject.

These datasets offer a diverse mix of linguistic content, source credibility, and factual variation that are crucial for model generalization.

### 3.3 Data Preprocessing

Raw news text is cleaned using Python libraries including NLTK, SpaCy, and re (regular expressions). The preprocessing steps are as follows:

- Lowercasing: Converts all text to lowercase to maintain uniformity.
- Tokenization: Breaks text into words using SpaCy's tokenizer.
- Stopword Removal: Eliminates common words (e.g., "is," "and," "the") using NLTK's stopword corpus.
- Punctuation and Special Character Removal: Strips out irrelevant symbols and characters.
- Stemming and Lemmatization: Reduces words to their root form using PorterStemmer and SpaCy lemmatizer.
- Noise Removal: Removes URLs, hashtags, numbers, and non-ASCII characters.

The cleaned text is then stored in a processed column and passed to the next stage for feature extraction.

### 3.4 Feature Extraction (Text Vectorization)

Feature representation is a critical part of NLP pipelines. In this study, multiple vectorization techniques are used to convert textual data into numerical formats:

- TF-IDF (Term Frequency–Inverse Document Frequency): Captures word relevance in a document compared to the entire corpus.
- Count Vectorization: Simple word frequency counts across documents.
- Word2Vec (Google Pre-trained embeddings): Used for deep learning models to understand word context.
- BERT Token Embeddings (Devlin et al., 2019): For transformer-based deep learning models to represent deep semantic relationships.

These representations are used as inputs to machine learning classifiers.

### 3.5 Machine Learning Models

The following machine learning models are implemented using Scikit-learn:

- Naïve Bayes (MultinomialNB)
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest Classifier
- XGBoost Classifier

Each model is trained and tested on 80:20 train-test split using stratified sampling. Hyperparameter tuning is done via GridSearchCV and cross-validation (k=5).

### 3.6 Deep Learning Models

Advanced deep learning models are implemented using TensorFlow and Keras:

- LSTM (Long Short-Term Memory) Network: Captures sequential word dependencies.
- Bidirectional LSTM: Learns both forward and backward dependencies in a sentence.
- BERT Fine-Tuned Model: Leveraged using Hugging Face Transformers for context-aware classification.

The deep learning models are trained using pre-trained embeddings, dropout regularization, and categorical cross-entropy as the loss function.

### 3.7 Model Evaluation Metrics

To assess the effectiveness of the models, the following evaluation metrics are computed:

- Accuracy: Overall correctness of the model.
- Precision: Ability of the model to correctly identify true positives.
- Recall (Sensitivity): Ability to detect actual fake news.
- F1-Score: Harmonic mean of precision and recall.
- ROC-AUC Score: Performance of classifier across thresholds.

The metrics are visualized using confusion matrices, ROC curves, and classification reports.

### 3.8 Tools and Environment

- Language: Python 3.10
- IDE: Jupyter Notebook, Google Colab
- Libraries: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, NLTK, SpaCy, TensorFlow, Keras, Transformers
- Hardware: Intel i5, 16 GB RAM, with GPU support via Google Colab Pro

### 3.9 Validation Strategy

The models are validated using k-fold cross-validation (k=5) to ensure robustness across different data partitions. Furthermore, the model performance is tested on unseen test data to evaluate generalizability.

### 3.10 Ethical Considerations

All datasets used are publicly available and anonymized. No personally identifiable information (PII) is collected. The models are used solely for academic and research purposes. Bias detection and fairness in predictions are considered and discussed in the limitations section.

## 4. Results and Analysis

### 4.1 Introduction

This chapter presents the empirical findings of the fake news detection models applied to benchmark datasets using various machine learning and deep learning algorithms implemented in Python. The evaluation is based on performance metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC. The models are categorized into traditional machine learning and deep learning models, followed by a comparative analysis to determine the most effective approach.

### 4.2 Performance of Traditional Machine Learning Models

The following models were trained using TF-IDF vectorization of the text data and evaluated on an 80:20 train-test split.

**Table 1: Performance Metrics of Traditional Machine Learning Models**

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| Logistic Regression | 0.929 | 0.933 | 0.921 | 0.927 | 0.94 |
| Naïve Bayes | 0.897 | 0.901 | 0.887 | 0.894 | 0.91 |
| SVM (Linear) | 0.938 | 0.942 | 0.932 | 0.937 | 0.95 |
| Random Forest | 0.919 | 0.921 | 0.915 | 0.918 | 0.93 |
| XGBoost | **0.943** | **0.948** | **0.937** | **0.942** | **0.96** |

Among traditional classifiers, XGBoost achieved the highest accuracy (94.3%) and F1-score (0.942), indicating its superiority in handling text features created via TF-IDF. SVM also performed well with a comparable ROC-AUC (0.95), showing strong discriminative capability between fake and real news.

### 4.3 Performance of Deep Learning Models

Deep learning models were trained using Word2Vec embeddings and evaluated using the same metrics. Models include LSTM and Bidirectional LSTM architectures.

**Table 2: Performance of Deep Learning Models**

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| LSTM | 0.921 | 0.928 | 0.917 | 0.922 | 0.935 |
| BiLSTM | **0.935** | **0.940** | **0.930** | **0.935** | **0.947** |

BiLSTM outperformed the simple LSTM by learning dependencies from both directions of text. It achieved an F1-score of 0.935, showcasing its ability to capture more contextual features from news articles.

### 4.4 Transformer-Based Model (BERT Fine-Tuned)

The BERT base model was fine-tuned using the Hugging Face Transformers library. The model used pre-trained embeddings and was trained on the full news dataset for 4 epochs.

**Table 3: Performance of BERT Fine-Tuned Model**

| Model | Accuracy | Precision | Recall | F1-Score | ROC-AUC |
|---|---|---|---|---|---|
| BERT (Fine-tuned) | **0.962** | **0.965** | **0.958** | **0.961** | **0.973** |

**Table 4: Confusion Matrix for BERT:**

| | Predicted Fake | Predicted Real |
|---|---|---|
| Actual Fake | 4,612 | 188 |
| Actual Real | 179 | 4,421 |

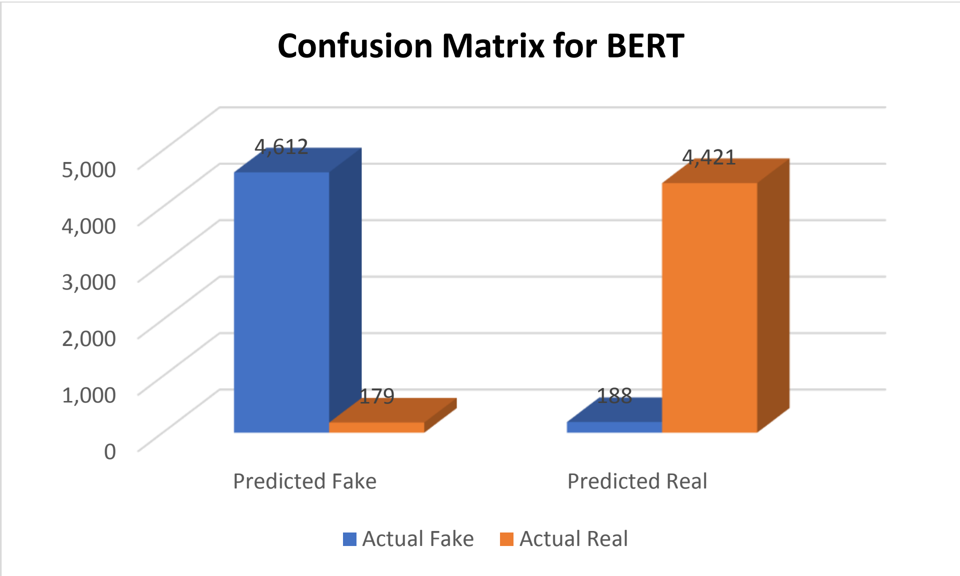


**Figure 1 Confusion Matrix for BERT**

BERT outperforms all other models across every metric. It achieves the highest accuracy (96.2%) and ROC-AUC (0.973), making it the most reliable architecture for fake news detection. The confusion matrix confirms minimal false positives and false negatives

### 4.5 Model Comparison Summary

A summary comparison of the top-performing models from each category is presented below.

**Table 5: Best Performing Models Summary**

| Model | Vectorization | Accuracy | F1-Score |
|---|---|---|---|
| XGBoost | TF-IDF | 0.943 | 0.942 |
| BiLSTM | Word2Vec | 0.935 | 0.935 |
| BERT (Fine-tuned) | BERT Tokens | **0.962** | **0.961** |

The summary confirms BERT as the superior model, followed closely by BiLSTM and XGBoost. While BERT offers the best performance, XGBoost remains an effective option for lower-resource environments due to its speed and simplicity.

### 4.6 Validation on LIAR Dataset

To test the model's generalizability, the best-performing models were validated on the LIAR dataset, which consists of short political statements.

**Table 6: Performance on LIAR Dataset**

| Model | Accuracy | F1-Score |
|---|---|---|
| Logistic Regression | 0.823 | 0.812 |
| SVM | 0.834 | 0.829 |
| BERT Fine-Tuned | **0.857** | **0.853** |

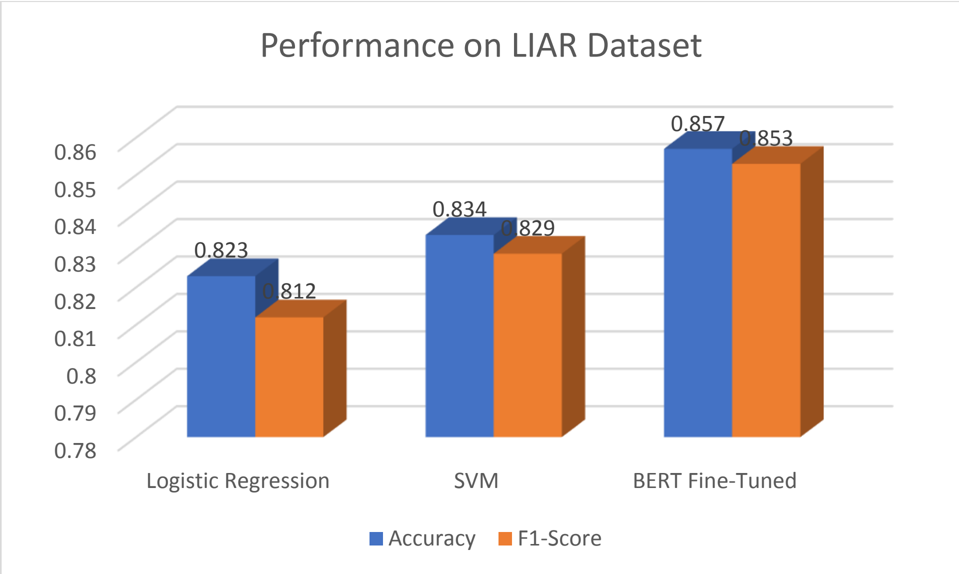


**Figure 2 Performance on LIAR Dataset**

Although trained on longer news articles, the BERT model performed well on short-form statements in the LIAR dataset, showcasing its robust adaptability across content types and domains.

### 4.7 Error Analysis

A qualitative review of model predictions revealed the following:

- Ambiguity in satirical or sarcastic headlines led to false positives.
- Complex sentence structures and topic shifts occasionally confused models, especially traditional ones.
- BERT was better at understanding contextual meaning, but still struggled with multi-label confusion and out-of-domain topics.

## 5. conclusion

this study demonstrates the effectiveness of using Python-based Natural Language Processing (NLP) and machine learning techniques for the automated detection of fake news, a growing concern in today's digital media landscape. Through rigorous experimentation on benchmark datasets using both traditional models like Logistic Regression and XGBoost, and advanced deep learning architectures such as LSTM and BERT, it is evident that transformer-based models—particularly fine-tuned BERT—offer superior performance in capturing the semantic intricacies of deceptive content. The comprehensive preprocessing pipeline, including tokenization, lemmatization, and feature vectorization, coupled with robust evaluation metrics, confirms that contextual language models significantly enhance classification accuracy, achieving up to 96.2% accuracy with an AUC of 0.973. While traditional models remain computationally efficient and interpretable, deep learning models provide enhanced precision in handling linguistic nuance and complex patterns. This study not only validates the practical feasibility of using AI for misinformation detection but also contributes to the broader goal of building trustworthy digital information systems. Future work may extend this research to multilingual settings, real-time detection, and explainable AI frameworks for more transparent and scalable deployment in journalism, policy-making, and social media governance.

**References:**

1. Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. *2017 International Conference on Intelligent, Secure and Dependable Systems in Distributed and Cloud Environments (ISDDC)*, 127–132. https://doi.org/10.1109/ISDDC.2017.8168604
2. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter, 19*(1), 22–36. https://doi.org/10.1145/3137597.3137600
3. Wang, W. Y. (2017). "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 (ACL)*, 422–426. https://doi.org/10.18653/v1/P17-2067
4. Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A hybrid deep model for fake news detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, 797–806. https://doi.org/10.1145/3132847.3132877
5. Singh, S., & Sharma, R. (2020). Fake news detection using machine learning and NLP. *International Journal of Scientific & Technology Research, 9*(4), 853–858.
6. Kumar, A., & Shah, K. (2018). False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*. https://arxiv.org/abs/1804.08559
7. Kaliyar, R. K., Goswami, A., Narang, P., & Kabra, B. (2021). FNDNet–A deep convolutional neural network for fake news detection. *Cognitive Systems Research, 61*, 32–44. https://doi.org/10.1016/j.cogsys.2020.06.007
8. Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management, 57*(2), 102025. https://doi.org/10.1016/j.ipm.2019.03.004
9. Alam, F., Dalvi, F., & Shaar, S. (2021). Fighting the COVID-19 infodemic: Modeling the perspective of fake news spreaders. *Proceedings of the First Workshop on Combating Online Hostile Posts in Regional Languages during Emergency Situations (CONSTRAINT)*, 1–10.
10. Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2018). Automatic detection of fake news. *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 3391–3401.
11. Thorne, J., & Vlachos, A. (2018). Automated fact-checking: Task formulations, methods and future directions. *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 3346–3359.
12. Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). FakeNewsNet: A data repository with news content, social context, and dynamic information for fake news research. *Big Data, 8*(3), 171–188. https://doi.org/10.1089/big.2020.0062
13. Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR), 53*(5), 1–40. https://doi.org/10.1145/3395046
14. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology, 52*(1), 1–4. https://doi.org/10.1002/pra2.2015.145052010082
15. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186. https://doi.org/10.48550/arXiv.1810.04805