International Journal of Computer Techniques–IJCT Volume 13 Issue 3, May-June - 2026

Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

AI-Based Fake News Detection System Using TF-IDF Vectorization and Logistic Regression

A. TharunTeja Reddy, B. Sairam Krishna, Ch. Abhiram Swarup

Department of Artificial Intelligence and Data Science

Dhanlakshmi Srinivasan University, Tiruchirapalli, India

Supervisor: ***Mr. K. Senthil Kumararaja***

Abstract — The rapid proliferation of misinformation across digital news platforms poses a significant threat to public discourse, democratic institutions, and societal well-being. With billions of users consuming information through social media and online portals, the distinction between credible journalism and fabricated content has become increasingly difficult. This paper presents a comprehensive AI-based Fake News Detection System that leverages Term Frequency-Inverse Document Frequency (TF-IDF) vectorization combined with a Logistic Regression classifier to automatically distinguish real news from fabricated content. The proposed system is trained and evaluated on a combined dataset of over 44,000 labeled news articles drawn from both reputable journalistic outlets and known misinformation sources. The complete pipeline incorporates a rigorous multi-stage text preprocessing module, TF-IDF feature extraction with corpus-level normalization, and a discriminatively trained Logistic Regression model with L2 regularization. Experimental results demonstrate an overall classification accuracy of 98.73%, with an F1-score of 0.99 achieved for both the real and fake news classes, confirming balanced detection performance. A detailed ablation study further validates each preprocessing step's contribution to final performance. The system additionally incorporates model persistence via Python's pickle module, enabling efficient real-time inference without retraining overhead. An interactive command-line interface (CLI) supports real-world deployment for end-user news verification.

Comparative benchmarking against Naive Bayes, Decision Tree, and Random Forest classifiers under identical experimental conditions demonstrates that Logistic Regression achieves the best accuracy-efficiency tradeoff.

Keywords — Fake News Detection; TF-IDF Vectorization; Logistic Regression; Natural Language Processing; Misinformation; Text Classification; Machine Learning; Feature Engineering; Binary Classification.

I. INTRODUCTION

The exponential growth of social media platforms and online publishing ecosystems has created an unprecedented information environment in which accurate journalism and fabricated misinformation coexist and compete for audience attention. Platforms such as Facebook, Twitter, and WhatsApp now serve as primary news sources for large segments of the global population, yet these platforms lack the editorial gatekeeping mechanisms of traditional media. As a result, deliberately fabricated stories — commonly referred to as "fake news" — can propagate virally before any fact-checking intervention takes place.

Empirical studies underscore the severity of the problem. Vosoughi et al. [1] demonstrated through a large-scale analysis of Twitter data that false information spreads approximately six times faster than true information, reaching broader audiences with significantly greater depth and breadth. The consequences of unchecked misinformation are multi-dimensional: in the public health domain, false medical claims have led to vaccine hesitancy and dangerous home remedies; in the political domain, fabricated electoral fraud narratives have eroded institutional trust; and in the financial domain, stock-market manipulating fake news has cost investors billions of dollars.

Manual fact-checking, while rigorous, cannot scale to match the volume and velocity of online content creation. Automated Natural Language Processing (NLP) and machine learning approaches therefore represent a critical technical intervention. Numerous methods have been explored in the literature, ranging from classical bag-of-words models with linear classifiers to deep neural networks leveraging contextual embeddings. Despite the performance gains offered by transformer-based architectures such as BERT [5], their computational demands render them impractical for deployment in resource-constrained environments, particularly in developing-world contexts or on edge devices.

This paper demonstrates that a carefully engineered classical machine learning pipeline — specifically TF-IDF vectorization combined with Logistic Regression — achieves near-state-of-the-art performance at a fraction of the computational cost. The complete system processes 44,919 news articles, achieves 98.73% classification accuracy, and classifies new articles in milliseconds on commodity hardware.

The primary contributions of this work are as follows:

(1) A complete end-to-end fake news detection pipeline encompassing data ingestion, multi-stage text preprocessing, TF-IDF feature extraction, Logistic Regression training, model persistence, and real-time CLI-based inference.

(2) An extensive empirical evaluation on a large-scale 44,919-article labeled dataset, achieving 98.73% accuracy and F1 = 0.99 for both classes.

(3) A systematic ablation study quantifying the contribution of individual preprocessing steps to overall classification performance.

(4) A comparative benchmark of four classical classifiers — Logistic Regression, Naive Bayes, Decision Tree, and Random Forest — under controlled experimental conditions.

(5) Discussion of system limitations and concrete directions for future enhancement, including semantic modeling and multilingual extension.

The remainder of this paper is organized as follows: Section II reviews related work; Section III describes the dataset; Section IV details the methodology; Section V presents experimental results; Section VI provides a comprehensive discussion; Section VII concludes with future directions.

II. LITERATURE REVIEW

The problem of automated fake news detection has attracted substantial research attention, particularly following high-profile misinformation incidents during the 2016 U.S. presidential election cycle. Shu et al. [2] provided a foundational taxonomy of detection approaches, grouping them into four categories: knowledge-based methods that check factual consistency against trusted knowledge graphs; style-based methods that analyze linguistic and rhetorical features; propagation-based methods that model

diffusion patterns over social networks; and source-based methods that evaluate publisher credibility.

A. Classical Machine Learning Approaches

Early detection systems relied on manually engineered lexical features combined with classical classifiers. Ahmed et al. [4] explored unigram and bigram TF-IDF representations with Logistic Regression and Linear SVM, achieving over 92% accuracy on benchmark corpora. Pérez-Rosas et al. [8] combined n-gram features, readability metrics, and psycholinguistic LIWC features, finding that surface-level linguistic cues reliably differentiate fabricated from authentic articles. These works establish TF-IDF as a robust, interpretable feature representation for the fake news domain.

Castillo et al. [9] proposed credibility assessment on Twitter by modeling social engagement features alongside content features. Their finding that the credibility of a tweet is strongly influenced by the retweet network topology motivated subsequent hybrid approaches that combine textual and network signals.

B. Deep Learning and Transformer Approaches

Recurrent architectures have been applied to capture sequential dependencies in news text. Ruchansky et al. [3] proposed the CSI (Capture, Score, Integrate) model, a hybrid deep framework that encodes article text with a recurrent network, models user response behavior, and integrates publisher source information, demonstrating that multi-modal signals improve over text-only baselines.

The introduction of BERT (Bidirectional Encoder Representations from Transformers) by Devlin et al. [5] marked a paradigm shift. BERT's pre-training on massive corpora followed by task-specific fine-tuning enabled contextual understanding of semantic nuances, irony, and implicit framing that escape surface-level lexical models. Kula et al. [10] fine-tuned BERT for Romanian fake news detection and achieved 98.6% accuracy, confirming transformer effectiveness across languages. However, BERT requires GPU acceleration and exhibits inference latencies orders of magnitude greater than classical models, limiting its deployment scope.

C. Dataset Contributions

Benchmark dataset availability has been a key driver of reproducible research. Wang [6] introduced the LIAR dataset, containing 12,836 hand-labeled short statements from PolitiFact spanning six truthfulness categories. FakeNewsNet [2] provides a richer corpus incorporating publisher social context and knowledge graph alignments. The dataset employed in this work, aggregating True.csv and Fake.csv labeled corpora, has become a widely adopted benchmark enabling direct performance comparisons across publications.

D. Positioning of This Work

While deep learning models achieve marginally higher performance ceilings, the literature consistently shows that classical TF-IDF + linear classifier pipelines remain highly competitive. Wang [6] and Ahmed et al. [4] both note that n-gram baselines provide strong lower bounds against which neural approaches must demonstrate meaningful improvement. The present work demonstrates that, through careful preprocessing engineering and systematic hyperparameter selection, Logistic Regression achieves 98.73% accuracy — within the margin of several transformer benchmarks — while training in under 5 seconds on a standard CPU.

III. DATASET

A. Data Sources and Composition

The dataset employed in this study comprises two complementary subsets sourced from publicly available labeled corpora. The first subset consists of 21,417 articles labeled as real news (class label 0), collected from reputable journalistic outlets including Reuters, The New York Times, The Guardian, and The Washington Post. Each article includes a title field and a full-body text field spanning multiple paragraphs.

The second subset contains 23,502 articles labeled as fake news (class label 1), aggregated from websites that have been independently flagged by professional fact-checking organizations (PolitiFact, Snopes) for systematic publication of fabricated or highly misleading

content. Topics within the fake news corpus span political conspiracy theories, health misinformation, sensationalist celebrity rumors, and fabricated crime reports.

The combined dataset contains 44,919 articles, yielding a near-balanced class distribution: 47.68% real and 52.32% fake. This near-balance is significant because it prevents classifier bias toward the majority class and allows accuracy to serve as a meaningful performance metric without requiring oversampling corrections.

TABLE I — DATASET COMPOSITION SUMMARY

Subset	Articles	Avg. Tokens	Class Label
Real News	21,417	~520	0
Fake News	23,502	~480	1
Combined	44,919	~500	—

B. Train-Test Split

The combined dataset was randomly shuffled (random seed = 42 for reproducibility) and partitioned using a 75/25 stratified split, yielding 33,689 training samples and 11,230 test samples. Stratification ensures that the class distribution is preserved in both splits, preventing accidental bias. This split follows the convention established in the comparative literature [4], enabling direct performance comparison.

C. Data Quality Observations

Manual inspection of the fake news corpus revealed several recurring linguistic characteristics that distinguish it from the real news corpus: (i) prevalence of ALL-CAPS emphasis for emotional amplification; (ii) extensive use of hyperbolic adjectives and superlatives; (iii) inclusion of URL references to fringe websites; (iv) fragmented HTML boilerplate embedded in article bodies; and (v) heavy use of passive attribution phrases ("sources say," "many people believe") without verifiable citations. These observations motivated the specific design choices in the preprocessing pipeline described in Section IV.

IV. METHODOLOGY

A. System Architecture Overview

The proposed fake news detection system follows a modular five-stage pipeline architecture: (1) Data Ingestion and Labeling; (2) Text Preprocessing; (3) TF-IDF Feature Extraction; (4) Logistic Regression Training; and (5) Model Persistence and CLI Inference. Each stage is described in detail in the following subsections. The modular design ensures that individual components can be independently upgraded — for instance, replacing the Logistic Regression classifier with a neural model — without requiring changes to the rest of the pipeline.

B. Text Preprocessing

Raw article text is formed by concatenating the title and body fields, separated by a single space. This concatenation is motivated by the observation that article titles in the fake news domain frequently contain the most sensationalist and discriminative linguistic content. The combined text then undergoes an eight-stage sequential cleaning pipeline:

Stage 1 — Lowercasing: All characters are converted to lowercase to normalize lexical variants. For example, "VACCINE" and "vaccine" are mapped to the same token, preventing artificial feature multiplication.

Stage 2 — Bracketed Text Removal: Regular expression pattern \[.*?\] is applied to strip editorial annotations, citation markers, and correction notes embedded within brackets. These tokens are meta-textual and do not carry topical discriminative information.

Stage 3 — Non-Word Character Removal: Characters that are not alphanumeric or whitespace are removed using \W+ substitution. This eliminates punctuation artifacts that would otherwise create low-frequency noise tokens.

Stage 4 — URL Stripping: Hyperlinks (both http and https protocols) are detected via regular expression and removed. URLs appear disproportionately in fake news content, but their inclusion as features would introduce domain-specific leakage rather than transferable topical signals.

Stage 5 — HTML Tag Removal: Residual HTML markup (paragraph tags, anchor tags, span elements) is stripped using a generic <.*?> pattern, correcting scraping artifacts in the raw corpus.

Stage 6 — Punctuation Deletion: Standalone punctuation tokens are removed after prior stages to ensure a clean alphanumeric token space.

Stage 7 — Alphanumeric Noise Removal: Tokens containing a mix of alphabetic and numeric characters (e.g., "abc123") that do not constitute meaningful English words are filtered, removing identifiers, reference codes, and scraping residuals.

Stage 8 — Newline and Whitespace Normalization: Embedded \n and \r characters are replaced with spaces, and multi-space sequences are collapsed to single spaces, yielding a clean, uniform token stream.

C. TF-IDF Vectorization

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical weighting scheme that assigns importance scores to terms based on their local frequency within a document balanced against their global rarity across the corpus. For a term t in document d within corpus D:

$$TF(t, d) = f(t, d) / \Sigma_{t' \in d} f(t', d)$$

$$IDF(t, D) = \log(|D| / |\{d \in D : t \in d\}|)$$

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

where f(t, d) is the raw frequency of term t in document d, |D| is the total number of documents, and the denominator of IDF is the number of documents containing t. Terms that appear in almost every document receive near-zero IDF weights, effectively functioning as data-driven stop words in addition to the explicit English stop word list applied.

The scikit-learn TfidfVectorizer is configured with the following parameters: (i) english stop word removal; (ii) max_df = 0.7, excluding terms appearing in more than 70% of documents to suppress near-universal tokens; (iii) sublinear TF scaling (log(1 + f)) to compress the dynamic range of high-frequency terms; and (iv) L2 normalization of the output row vectors. Critically, the vectorizer is fit exclusively on the training set and then applied as a frozen transform to the test set, ensuring zero data leakage.

D. Logistic Regression Classifier

Logistic Regression is selected as the classifier for its strong theoretical guarantees, efficiency on high-dimensional sparse feature spaces, and natural probability output. The binary prediction model is defined as:

$$P(y = 1 \mid x) = \sigma(w \cdot x + b) = 1 / (1 + \exp(-(w \cdot x + b)))$$

where x ∈ ℝ^d is the TF-IDF feature vector, w ∈ ℝ^d is the weight vector, and b is the bias term. The decision boundary P(y=1|x) = 0.5 corresponds to w·x + b = 0. The model is trained by minimizing the regularized cross-entropy loss:

$$L(w) = -(1/n) \Sigma_i [y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)] + (\lambda/2) ||w||^2$$

where the L2 regularization term (λ/2)||w||² penalizes large weight magnitudes, controlling overfitting on the high-dimensional TF-IDF space. The regularization strength hyperparameter C = 1/λ is set to its default value of 1.0. Optimization is performed via the limited-memory BFGS (lbfgs) solver, which exploits the convexity of the objective and converges reliably without requiring a learning rate schedule.

E. Model Persistence and Inference

Following training, both the TfidfVectorizer and the LogisticRegression model objects are serialized to disk using Python's pickle module. This persistence mechanism decouples the training phase from inference, enabling new articles to be classified without reloading or reprocessing the training data. The inference pipeline operates as follows: (1) user-supplied raw text is fed into the preprocessing function; (2) the cleaned text is transformed by the saved vectorizer into a TF-IDF feature vector; (3) the saved Logistic Regression model applies the learned weight vector to produce a probability score; and (4) the probability is thresholded at 0.5 to yield a binary label displayed to the user.

F. Interactive CLI Interface

The command-line interface provides real-time news verification for end users. Upon launching the application, the pre-trained model

and vectorizer are loaded from disk. The interface enters a loop prompting the user to paste or type article text. Upon submission, the classification result is printed as ' Real News' or ' Fake News' alongside a confidence probability score. The session terminates gracefully when the user inputs 'exit'. The entire inference cycle, from input submission to result display, completes in under 50 milliseconds on a standard laptop CPU.

V. EXPERIMENTAL RESULTS

A. Evaluation Metrics

Classification performance is evaluated using the standard metrics for binary classification: Precision, Recall, F1-Score, and Overall Accuracy. Precision measures the fraction of predicted positive instances that are truly positive; Recall measures the fraction of actual positive instances that are correctly retrieved; and F1-Score is their harmonic mean, providing a balanced measure that accounts for both false positives and false negatives.

B. Classification Performance

Table II presents the full per-class classification report generated on the held-out test set of 11,230 articles.

TABLE II — CLASSIFICATION REPORT ON TEST SET

Class	Precision	Recall	F1-Score	Support
Real (0)	0.99	0.98	0.99	5,354
Fake (1)	0.98	0.99	0.99	5,876
Accuracy	—	—	0.9873	11,230
Macro Avg	0.99	0.99	0.99	11,230
Wtd. Avg	0.99	0.99	0.99	11,230

The system achieves an overall accuracy of 98.73% on the 11,230-article test set. Both classes achieve an F1-score of 0.99, indicating near-perfect balanced performance. The Real News class achieves precision of 0.99 and recall of 0.98, indicating that 1% of real articles are incorrectly flagged as fake (false positives), while 2% of fake articles are missed (false negatives for the fake class). The Fake News class achieves recall of 0.99, meaning the system correctly identifies 99% of fabricated articles — a critical property for practical deployment where missing fake news (false negatives) carries the highest societal cost.

C. Ablation Study

To quantify the contribution of individual preprocessing steps, an ablation experiment was conducted by progressively removing stages from the pipeline. Table III reports accuracy with each step omitted.

TABLE III — ABLATION STUDY: EFFECT OF PREPROCESSING STEPS

Configuration	Accuracy (%)
Full pipeline (all steps)	98.73
Without URL stripping	98.41
Without HTML removal	98.29
Without bracketed text removal	98.55
Without lowercasing	97.92
Without stop word removal	97.60
Raw text only (no preprocessing)	95.34

The ablation results confirm that every preprocessing step contributes positively to final accuracy. The greatest single-step contribution comes from stop word removal (+1.13% over no-preprocessing baseline), followed by lowercasing (+0.81%). The combined pipeline delivers a 3.39% absolute accuracy gain over completely unprocessed text, underscoring the importance of domain-informed feature engineering.

D. Comparative Classifier Benchmark

Table IV compares Logistic Regression against three alternative classical classifiers trained under identical dataset splits and preprocessing conditions.

TABLE IV — COMPARATIVE MODEL PERFORMANCE

Model	Accuracy (%)	F1-Score	Train Time
Logistic Regression (Ours)	98.73	0.99	~4 sec
Random Forest	97.30	0.97	~45 sec
Naive Bayes	94.21	0.94	~2 sec
Decision Tree	89.67	0.90	~6 sec

Logistic Regression achieves the highest accuracy (98.73%) and the best F1-score (0.99) among all tested classifiers. Random Forest follows at 97.30% but requires approximately 45 seconds of training time — more than 11× slower than Logistic Regression — due to the overhead of training 100 individual decision trees. Naive Bayes, though the fastest model (~2 seconds), achieves 94.21% accuracy, trailing by 4.52 percentage points. The Decision Tree classifier achieves the lowest accuracy (89.67%), likely due to overfitting on the high-dimensional TF-IDF feature space without ensemble regularization. These results confirm that the discriminative linear model is the optimal choice for this high-dimensional sparse text classification task.

VI. DISCUSSION

A. Why TF-IDF Succeeds for Fake News Detection

The strong performance of TF-IDF features on this task can be attributed to systematic vocabulary differences between authentic and fabricated news. Fake news articles exploit a recognizable lexical signature: hyperbolic adjectives ("explosive," "unbelievable," "shocking"), polarizing political terms, emotionally charged verbs, and sensationalist framing devices. TF-IDF assigns high weights to these discriminative terms because they appear frequently within the fake news corpus but are suppressed in the real news corpus, producing feature vectors that are highly separable in the logistic regression decision space.

Conversely, real news articles feature measured, citation-grounded language, institutional source attributions, and balanced hedging terminology. The max_df = 0.7 threshold plays an important role in excluding generic journalistic terms shared by both classes, sharpening the discriminative power of the remaining features.

B. Error Analysis

The 1.27% misclassification rate (142 articles) warrants examination. Manual inspection of false positives (real articles classified as fake) reveals that genuine investigative journalism articles employing dramatic narrative framing and emotionally charged language sometimes exhibit lexical profiles similar to fabricated content. False negatives (fake articles classified as real) tend to occur for professionally written propaganda pieces that mimic journalistic style, including measured tone, institutional citations, and source attribution, without the typical lexical markers of low-quality misinformation.

These error patterns highlight a fundamental limitation of surface-level lexical modeling: adversarial fake news that deliberately adopts authentic journalistic vocabulary will evade detection. This motivates the incorporation of semantic, structural, and provenance-based features in future iterations.

C. Computational Efficiency and Deployment Considerations

A major practical advantage of the proposed system is its computational efficiency. Training completes in approximately 4 seconds on a standard dual-core laptop CPU (Intel Core i5, 8 GB RAM), and inference for a single article requires less than 50 milliseconds. The serialized model pair occupies less than 15 MB on disk. These characteristics make the system deployable in environments where GPU acceleration is unavailable, including low-cost cloud instances, institutional servers in developing countries, and browser-based WebAssembly sandboxes.

D. Comparison with Deep Learning Baselines

While BERT-based models achieve marginally higher accuracy on some benchmark datasets, the gap relative to our system is narrow.

Kula et al. [10] report 98.6% accuracy with fine-tuned BERT — only 0.13% above our Logistic Regression baseline — while requiring GPU hardware and inference latencies of 200–500 ms per article. For applications where millisecond-level inference is critical, such as browser extension-based real-time verification, our classical pipeline offers a substantially more viable deployment profile.

E. Limitations

Several limitations of the current system should be acknowledged. First, the dataset is composed exclusively of English-language articles; performance on non-English content is unknown. Second, the system relies on purely lexical features and has no access to publisher metadata, social engagement signals, or temporal context. Third, the labeled dataset, while large, reflects the fake news patterns of a specific time period; concept drift over time may degrade performance as misinformation strategies evolve. Fourth, the system is not adversarially robust — a sophisticated attacker with knowledge of the model's vocabulary weights could craft fake articles that evade detection by substituting flagged terms with semantically equivalent but low-weight alternatives.

VII. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive end-to-end AI-based Fake News Detection System employing TF-IDF vectorization and Logistic Regression. The system was rigorously evaluated on a 44,919-article labeled dataset, achieving 98.73% overall accuracy with an F1-score of 0.99 for both the real and fake news classes. Ablation experiments validated the contribution of each preprocessing stage, and comparative benchmarking confirmed that Logistic Regression provides the optimal accuracy-efficiency tradeoff among classical classifiers. The system's computational efficiency — training in ~4 seconds, inference in <50 ms — and its persistence architecture make it immediately deployable for real-world fake news verification applications.

The following directions are proposed for future work:

(1) Semantic Feature Integration: Incorporation of word embeddings (Word2Vec, GloVe) and contextual sentence embeddings (Sentence-BERT) to capture semantic rather than purely lexical signals, improving robustness against adversarial vocabulary substitution.

(2) Transformer Fine-Tuning: Fine-tuning of lightweight transformer models (DistilBERT, ALBERT) to explore the performance ceiling achievable with contextual embeddings while managing computational overhead.

(3) Multilingual Extension: Expansion of the training corpus to include Hindi, Telugu, Tamil, and other Indic languages to address misinformation in regional digital media ecosystems.

(4) Multi-Modal Feature Fusion: Integration of publisher credibility scores, social engagement metrics (shares, reactions), image forensics features, and knowledge-graph consistency checks into a unified multi-modal classifier.

(5) Web API and Browser Extension Deployment: Development of a REST API and accompanying browser extension that provides real-time fake news classification for any article a user encounters while browsing, incorporating the model's confidence score as an overlay annotation.

(6) Adversarial Robustness: Investigation of adversarial training techniques and certified defenses to harden the classifier against motivated adversaries who attempt to craft fake news articles that evade detection.

ACKNOWLEDGMENT

The authors sincerely thank Mr. K. Senthil Kumararaja, Assistant Professor, Department of Artificial Intelligence and Data Science, Dhanlakshmi Srinivasan University, Tiruchirapalli, for his invaluable guidance, constructive feedback, and continuous supervision throughout this research project. The authors also gratefully acknowledge the open-source community for providing the labeled datasets and the scikit-learn, pandas, and NLTK libraries that form the technical foundation of this work. Special thanks are extended to the reviewers whose detailed feedback strengthened this manuscript.

REFERENCES

[1] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018.

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

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

[4] H. Ahmed, I. Traore, and S. Saad, "Detection of online fake news using N-gram analysis and machine learning techniques," in Proc. ISDDC, 2017, pp. 127–138.

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

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

[7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[8] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in Proc. COLING, 2018, pp. 3391–3401.

[9] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in Proc. WWW, 2011, pp. 675–684.

[10] S. Kula, M. Choraś, R. Kozik, P. Ksieniewicz, and M. Woźniak, "Sentiment analysis for fake news detection by means of neural networks," in Proc. ICCS, 2020, pp. 653–666.

[11] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," ACM Comput. Surv., vol. 51, no. 2, pp. 1–36, 2018.

[12] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," in Proc. UKRPROG, 2017, pp. 1–4.