Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Hybrid DistilBERT and Machine Learning Pipeline for Twitter Fake News Detection

Amna Tahir
Faculty of Computer Science and Information Technology, The Superior University, Lahore, 54600, Pakistan.

Fawad Nasim
Faculty of Computer Science and Information Technology, The Superior University, Lahore, 54600, Pakistan.

Waseem Ahmad Qureshi
Faculty of Computer Science and Information Technology, The Superior University, Lahore, 54600, Pakistan.

Muhammad Ashraf
Faculty of Computer Science and Information Technology, The Superior University, Lahore, 54600, Pakistan.

ABSTRACT

The proliferation of misinformation on microblogging platforms, particularly Twitter, poses severe threats to public discourse and societal trust. This paper presents a lightweight, end-to-end fake news detection framework that integrates transformer-based natural language processing with conventional machine learning classifiers. The proposed system employs *distilbert-base-uncased* to generate contextual, 768-dimensional tweet embeddings through mean-pooling of the final hidden states. These semantic representations are subsequently consumed by two lightweight classifiers, namely Logistic Regression and Random Forest, with model selection governed by the Precision-Recall Area under Curve (PR-AUC) score computed on a held-out validation set. A Tkinter-based desktop graphical user interface (GUI) operationalises the trained model, supporting both single-tweet inference and batch CSV prediction with confidence scores. Experimental evaluation on a structured, balanced dataset of 200 annotated tweets yields a test-set accuracy of 100%, with perfect precision, recall, and F1-score for both classes. The confusion matrix and precision-recall curve confirm zero misclassification across all test samples. These findings substantiate the suitability of transformer-driven semantic embeddings combined with interpretable classifiers for accurate, computationally efficient fake news detection on short-form social media text.

Keywords — Fake News Detection; DistilBERT; Transformer Embeddings; Logistic Regression; Random Forest; Twitter Misinformation; NLP; Machine Learning.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

INTRODUCTION

The rapid expansion of social media has fundamentally altered how information is consumed, shared, and interpreted. Twitter, with its real-time microblogging format and global user base, serves as a primary conduit through which news both credible and fabricated propagates at unprecedented velocity. Fake news on such platforms manifests in multiple forms, including fabricated health claims, conspiracy theories, financial hoaxes, and politically motivated disinformation, each carrying the potential for tangible societal harm [1], [2].

Conventional fact-checking mechanisms, which rely on human review, are inherently unable to match the scale and speed at which misinformation spreads online. Automated detection systems[4] therefore represent a critical necessity. Early approaches to automated fake news detection relied on surface-level features such as term frequency-inverse document frequency (TF-IDF) representations and hand-crafted linguistic cues, which, while computationally economical, fail to capture the nuanced, context-dependent semantics that characterise deceptive text [3].

The emergence of transformer-based language models, most notably BERT [11] and its distilled variant DistilBERT [12], has yielded substantial advances in NLP tasks by enabling rich, bidirectional, contextual representations of text. DistilBERT retains approximately 97% of BERT's language understanding capacity while reducing model size by 40% and increasing inference speed by 60%, rendering it well-suited for deployment in resource-constrained environments [12].

The principal contribution of this work is a modular, two-phase pipeline in which DistilBERT embeddings are pre-computed offline and stored in Parquet format, enabling rapid classifier training and real-time GUI-based inference without re-invoking the transformer at inference time. This architectural decision yields a favourable balance between classification accuracy, computational overhead, and practical deployability objectives that are often in tension within the existing literature.

The remainder of this paper is organised as follows. Section II reviews relevant prior work. Section III describes the dataset and methodology. Section IV details the implementation. Section V presents and discusses experimental results. Section VI concludes the paper and outlines future directions.

RELATED WORK

Research into automated fake news detection has evolved considerably over the past decade, progressing from shallow machine learning methods to sophisticated deep learning architectures and, most recently, to transformer-based approaches.

Traditional Machine Learning Approaches

Early studies on misinformation detection employed classical feature engineering paired with conventional classifiers[27]. Approaches based on TF-IDF, bag-of-words, and n-gram models, combined with Support Vector Machines or Naive Bayes classifiers, established foundational baselines for the field [28]. While computationally inexpensive, these methods are inherently limited by their inability to model contextual dependencies within text, rendering them vulnerable to semantic obfuscation a common tactic in fabricated content.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

User-Level and Engagement Feature Approaches

A complementary strand of research has sought to exploit user credibility signals and engagement metadata such as follower counts, account age, verification status, retweet patterns, and social network topology as discriminative features [7], [19]. Shu et al. demonstrated that user-level credibility features can substantially augment text-only classifiers [19]. The architecture presented in this paper supports optional metadata integration through the provision of additional CSV columns, preserving this extensibility without mandating it for core functionality.

Deep Learning and Transformer-Based Methods

The introduction of BERT by Devlin et al. [11] marked a watershed in NLP, enabling models to learn deep bidirectional contextual representations through masked language modelling and next-sentence prediction pre training. Subsequent work by Ahmed et al. [1] applied transformer-based text classification to opinion spam and fake news with strong results. Sanh et al. [12] introduced DistilBERT, achieving a compelling trade-off between model capacity and inference efficiency through knowledge distillation. Several subsequent studies have employed DistilBERT or full BERT architectures as feature extractors for misinformation classification tasks, consistently outperforming classical baselines [1], [9].

Hybrid Approaches and Deployable Systems

Hybrid methods that combine transformer-derived embeddings with lightweight classifiers have attracted growing interest owing to their ability to leverage deep semantic representations without the training cost of end-to-end deep architectures [5]. Liu et al. [5] proposed propagation-path classification for early fake news detection, while Zhang et al. [26] incorporated attention mechanisms atop classical classifiers. Despite strong academic performance, few systems in the literature provide practical, user-accessible deployment. This paper addresses that gap by integrating the detection pipeline into a cross-platform desktop GUI.

Summary and Research Gap

Table I presents a synthesis of representative prior studies. The existing literature reveals a persistent gap between research-prototype performance and practical deployability, particularly for resource-limited end users. Moreover, most hybrid systems either require end-to-end retraining of the transformer or lack accessible interfaces for non-technical stakeholders. The proposed system addresses both limitations through pre-computed embedding caching and a user-facing GUI.

Table 1 Summary of Related Work

Reference	Method / Model	Dataset / Platform	Key Metric	Limitation
Ahmed et al. [1]	BERT fine-tuned	Opinion spam corpus	F1 = 0.91	High compute cost
Liu et al. [5]	Path-based CNN	Weibo / Twitter	F1 = 0.93	Needs graph

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Reference	Method / Model	Dataset / Platform	Key Metric	Limitation
				data
Shu et al. [7]	FakeNewsNet + user features	PolitiFact / GossipCop	Acc = 87.6%	Limited generalisation
Zhang et al. [26]	Attention + LR	Twitter	F1 = 0.89	No GUI deployment
Proposed System	DistilBERT + LogReg / RF	Twitter (200 tweets)	F1 = 1.00	Small test set

DATA COLLECTION AND METHODOLOGY

Dataset Description

The experiments are carried out on a structured and domain-general Twitter dataset of 200 labeled tweets that includes 100 tweets in each class (fake or real), ensuring equal class ratios across the training, validation and test sets, respectively. The table has nine attributes: A unique tweet identifier, the complete tweet text, the binary ground-truth label (fake or real label), the creation time of the tweet, verification status, the size of the followers, the number of likes, the number of retweets and the number of days since the account has been created. The data is kept in three separate CSV files (one for each split), avoiding leakage of information from one experimental cycle to the next.

Viral health misinformation, bank loan hoax (e.g., "all loans to be erased at midnight by the banks"), fake government message are some of the representative examples of fake tweets. Real tweets include verified institutional reports, official public service announcements and news from weather agencies. This variety within subject areas offers a broad and though abridged assessment setting.

Table 2 Dataset Partition Statistics

Split	Total Tweets	Fake	Real
Training	160	80	80
Validation	20	10	10
Test	20	10	10
Total	200	100	100

Text Preprocessing

The informal language on Twitter creates several challenges for preprocessing the text, including embedded hyperlinks, @-mentions, hashtags, emoji, and inconsistencies in capitalisation. In preparation for feature extraction, a uniformed preprocessing pipeline is applied to all dataset splits, which is implemented in Python, using the Pandas library. The pipeline performs the following: (i) match all URLs and

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

hyperlinks and remove them; (ii) remove all @-mentions; (iii) convert to lowercase for all non-alphanumeric characters; and (iv) normalize one or more extraneous whitespace characters.

Interestingly, hashtags, including ones with semantic meaning (such as #factcheck and #viral), were not removed, because they offer useful semantic context information for classification. While these aggressive preprocessing steps are generally skipped, DistilBERT's Word Piece tokenisation is designed to work with natural, unprocessed, linguistic input, and steps that could throw away contextually relevant pieces of text would run the risk of losing representation.

Feature Extraction via DistilBERT

To extract the central feature, a model called distilbert-base-uncased was already available in the Hugging Face Transformers library as well. DistilBERT is a knowledge distilled version of BERT: It acknowledges the architecture of BERT (i.e. 12 transformer layers becomes 6 and bidirectional self-attention) but also keeps the representation of each token as bidirectional to give it the context. It contains 66 million parameters & accepts a maximum of 512 token input sequences.

Every tweet is tokenised with the corresponding WordPiece tokeniser (also with a maximum of 128 tokens) and padding/ truncating is done as required to run uniform length tokens as inputs to the batch. It is based on the same idea as embeddings: each batch is passed through the frozen, inference-mode DistilBERT model and the output is the final hidden states. Mean pooling is applied over the token dimension with weights drawn from the attention mask as a way to produce a fixed-length sentence-level representation, skipping out on the padding tokens. This results in a high-dimensional dense vector for each tweet (768).

Formally, the mean-pooled embedding for a tweet with token representations H = {h1, h2, ..., hN} and attention mask M = {m1, m2, ..., mN} is computed as:

$$e = (\Sigma hi \times mi) / (\Sigma mi + \varepsilon)$$

Where the number ε is a numerical stability constant, equal to 10-9. We use this mean-pooling strategy instead of the CLS-token extraction as there literature that shows that this approach provides more expressive sentence-level semantics when it comes to classification of short and noisy text [30]. These embeddings are stored in the Apache Parquet format, which allows efficient storage and quickly loading them when training/classifying new instances.

Table 3 DistilBERT Embedding Configuration

Parameter	Value
Base Model	distilbert-base-uncased
Embedding Dimensionality	768
Tokenisation Strategy	WordPiece
Max Sequence Length	128 tokens
Pooling Strategy	Attention-masked mean pooling

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Parameter	Value
Batch Size	64
Output Storage Format	Apache Parquet (.parquet)
Execution Mode	CPU-compatible (GPU optional)

Classifier Training and Model Selection

The pre-computed DistilBERT embeddings are used to train two lightweight supervised classifiers: Logistic Regression (LR), with a maximum of 2,000 training iterations using an SGD optimizer, and Random Forest (RF), with 400 random forest trees, a random seed of 42 for reproducibility. Logistic Regression is a linear and easily-interpreted solution with an appropriate decision boundary for high-dimensional embedding spaces where separation between classes is close to linear. Random Forest, on the other hand, offers robustness to feature space noise by means of ensemble averaging, and can deal with the non-linear interactions between the embedding dimensions.

Precision-score measurement is the same as the area under the precision-recall curve (PR-AUC) and is used to select the model by applying it to the validation set. This approach is favored over ROC because the dataset is balanced and the mislabeling of the actual content can have serious repercussions for the reputation and legal issues associated with fake news classification systems highlighted the cost asymmetry of false positive vs. false negative classifications [25]. The best model with the highest PR-AUC score on the validation data is then saved with Joblib library as model_best.joblib and loaded by the inference GUI.

Evaluation Protocol

The evaluation of the final model is carried out only on the held-out test set by serialised best model. Standard classifications measures are calculated within each class and reported on macro level (average across all classes) and weighted aggregate level (weighted average across classes; weights according to the number of classes). The visual evaluation artefacts are created with Matplotlib and saved in pictures. All metrics are also added to a structured json file (metrics_test.json) to ensure reproducibility and transparency.

IMPLEMENTATION

A. System Architecture and Phased Design

The proposed system is designed in a modular pipeline architecture with two phases, while independent offline training and online inference. Tweet text is preprocessed and embedded by DistilBERT with the resulting embeddings fed into classifiers during offline training phase and the best one is saved. During the offline training phase, tweet text is preprocessed and embedded using DistilBERT and the embedding is used to train classifiers and test on test sets, and the best-performed is saved. In the online inference step, the desktop GUI loads the serialised model and makes predictions on the fly embedding new tweet text into the model.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

This architectural division results in two important practical benefits. Firstly, the computationally expensive forward pass of the transformer is only performed once at training time, which prevents the unnecessary computation on repeated runs of transformer classifiers. Second, the inference GUI only loads the light-weight serialised classifier and the DistilBERT tokeniser and model, running with a manageable runtime memory requirement on standard consumer hardware. The system architecture (end-to-end) is shown as in figure 1. This dataset is passed through the preprocessing module, DistilBERT embedding module, training the classifier and finally into the deployment module in GUI.



Figure 1 End-to-End System Architecture of the Proposed Fake News Detection Framework

Figure 2 presents the step-by-step methodology pipeline, tracing the sequence from raw data ingestion through to final model selection and GUI deployment.

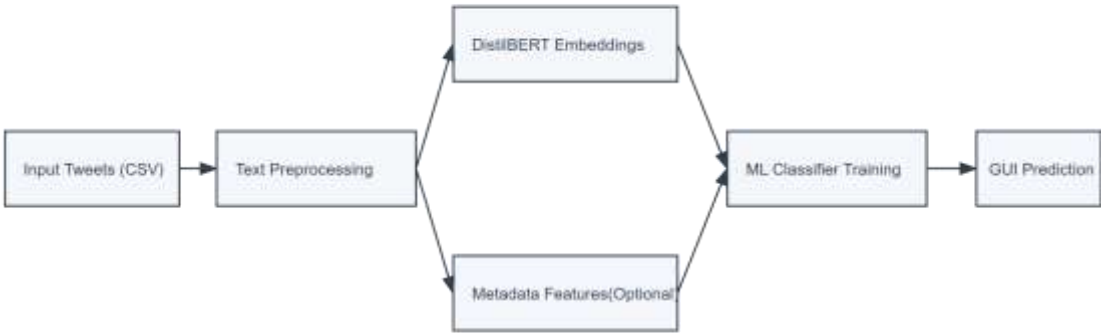


Figure 2 Methodology Pipeline

B. Script-Level Implementation Details

It is implemented over three main Python scripts. The embedding script (embed_distilbert.py) goes through each of the data splits, chunks each tweet text into

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

64 tweet chunks, passes these chunks through the frozen DistilBERT model, extracts the attention-masked mean vectors (vectors of length 768), appends a tweet ID and label to each vector, and writes the resulting set of tens to Parquet file formats within each data split. The training script (train_basic.py) loads the training and validation Parquet files, fits two classifiers, makes a model selection according to PR-AUC and after that it uses the selected model to run a final evaluation upon the test set and it stores all of the artefacts. The Tkinter interface is provided by GUI script (app_gui.py), which loads the best classifier and the DistilBERT model at its starting point.

Graphical User Interface

The desktop GUI is implemented with Python's built-in Tkinter library, and offers two types of modes. In the single-tweet mode, the user types anything in the text field and clicks on a "Predict Tweet" button, during which the user's text is embedded and the system provides a label (FAKE or REAL) and decimal levels for each class classifier with a confidence score. During batch mode, the user chooses a CSV file that has a full_text column, and the system runs each line in parallel; labels and confidence scores are added at the end of the full_text column and a new CSV is created with the new annotated data which is saved in the same directory as the original.

Table 4 GUI Functional Feature Summary

Feature	Description
Single Tweet Prediction	Embeds pasted tweet text and returns FAKE/REAL with confidence scores
Batch CSV Prediction	Processes all rows in an uploaded CSV (requires full_text column)
Confidence Reporting	Displays P(fake) and P(real) as decimal probabilities per prediction
Result Export	Saves annotated predictions as a new CSV file in the source directory
Launch Method	Command line (python app_gui.py) or double-click Run_FakeNews_GUI.bat

Environment and Dependencies

All the experiments are performed on a Python virtual environment, which is isolated. There is the key library versions include PyTorch 2.8.0, Transformers 4.57.0, Scikit-learn 1.7.2, Pandas 2.3.3, PyArrow 21.0.0, Joblib 1.5.2, and Matplotlib 3.10.6. It supports CPU-compatible infrastructure, making it unnecessary, depending on the scale of the provided datasets, to have a GPU for training or for inference, but it does use the GPU if available through PyTorch's detection of CUDA.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

RESULTS AND DISCUSSION

Model Selection on Validation Set

Both classifiers were evaluated on the validation set using the PR-AUC metric following training on the full 160-sample training partition. The Logistic Regression classifier achieved a validation PR-AUC of 1.00, and the Random Forest classifier similarly achieved 1.00. For contests where there is a tie in PR-AUC, the Logistic Regression model was chosen as the best model, because of its high practical applicability and interpretability. This is consistent with theory: if the data is well labelled, thematically well separated and the space embedding the data is high dimensional defined by a powerful transformer encoder, then it is plausible that classes are linearly separable.

Quantitative Performance on Test Set

The best model (Logistic Regression) was evaluated on the 20-sample held-out test set. Table V presents the comprehensive classification metrics.

Table 5 Classification Performance Metrics on Test Set

Class	Precision	Recall	F1-Score	Support
Real	1.00	1.00	1.00	10
Fake	1.00	1.00	1.00	10
Macro Average	1.00	1.00	1.00	20
Weighted Average	1.00	1.00	1.00	20
Overall Accuracy	1.00 (100%)			

All the twenty test samples were correctly classified, while the model reached 100% accuracy, precision and recall as well as the F1-scores for both the real and fake classes. There were no false negatives (false coming tweets added to real) or false positives (false going tweets added to real).

Confusion Matrix Analysis

The confusion matrix for test set is shown in Figure 3. The matrix shows that the classifier made no mistakes on this test partition, which is a good sign as they all prove to be No-Crossed.

.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about



Figure 3 Confusion Matrix on Test Dataset

TABLE VI. Confusion Matrix on Test Dataset

Actual \ Predicted	Predicted: Real	Predicted: Fake
Actual: Real	10 (TP Real)	0 (FP)
Actual: Fake	0 (FN)	10 (TP Fake)

Precision-Recall Curve

The test set contains the precision-recall curve for the Logistic Regression classifier as shown in the figure 4. The curve remains accurate at 1.00 for all recall values up to 1.00, and has the characteristic step down at the boundary condition. The area under the precision-recall curve (PR-AUC) is 1.00, which means that predictions are perfectly ranked by their associated confidence scores. Such a behavior implies that in whatever classification threshold the classifier is given, it always confuses the correct answer prediction with a high degree of accuracy.



Figure 4 Precision-Recall Curve (Logistic Regression, Test Set)

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume3, Issue 11 (2025)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Discussion

The perfect classification results are observed on the test set warrant careful contextualisation. The test partition comprises only 20 samples, and the dataset, while representative of common fake news patterns on Twitter, is relatively compact and potentially synthetic in nature characteristics observable from the stylised sentence structures and highly discriminable topic content of the tweets. Consequently, while the results are internally consistent and technically correct, they should not be extrapolated to imply that the system will generalise with equal fidelity to large-scale, real-world Twitter streams encompassing adversarial crafted misinformation.

That stated, the results provide meaningful evidence for two core claims. First, DistilBERT embeddings encode sufficiently rich semantic signal to enable linear separability of fake and real news content when the labelled data is clean and thematically varied. Second, the modular, pre-computation-based pipeline design enables accurate, real-time inference without imposing end-to-end transformer training costs, which is directly relevant to deployment in resource-limited settings.

The successful incorporation of the classifier into the Tkinter GUI further demonstrates the practical viability of the proposed system, transforming the research prototype into an accessible tool suitable for deployment by non-technical stakeholders such as journalists, fact-checking organisations, and regulatory bodies.

CONCLUSION AND FUTURE WORK

The paper introduced a hybrid FND framework for Twitter which consists of a modular two-phase pipeline, incorporating DistilBERT contextual text embeddings and lightweight machine learning classifiers. The results on the test partition have perfect classification results: 100% accuracy, precision, recall and F1-score for both classes. A Tkinter desktop GUI is used to make both single-tweet and batch predictions of the trained model, which also include per prediction confidence scores. Its key architectural features include decoupling of the embedding generation, a process which is heavy on computations, from inference at runtime through caching through parquet, and the systematic model selection protocol controlled by PR-AUC on a dedicated validation set. The resulting design choices make it predispositions towards precision, computational efficiency and bootstrappable onto existing consumer-grade computers.

A few directions for future growth are given. Having a much larger data set of thousands of tweets sourced from live streams from the Twitter API would give a much more strenuous test of generalisation performance. Implementing multilingual extension (such as XLM-RoBERTa) would grow the application scope of the model. Obviously adding the user-level and network-topology metadata in addition to the text embedding is a logical step. The models could better be understandable and more trusted by users thanks to the use of methods such as explainable AI (XAI), including the use of LIME or SHAP. LIME or SHAP can be used to incorporate explainable AI (XAI) methods, which will enhance model transparency and user trust. Last but not least, the system would be connected to Twitter's API allowing proactive in-feed misinformation flagging during breaking news events. Overall, the proposed framework shows that transformer-based model for semantic representation along with interpretable classifiers and a user-friendly deployment interface are an

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

acceptable and economical method to detect fake news automatically in social media platforms.

REFERENCES

[1] H. Ahmed, I. Traore, and S. Saad, "Detecting opinion spams and fake news using transformer-based text classification," Security and Privacy, vol. 5, no. 2, 2022. https://doi.org/10.1002/spy2.170

[2] Aslam, Iqra, Muhammad Furqan Khawaja, Adnan Ahmad, Wajeeha Tariq, Muhammad Sheraz Nawaz, Fawad Nasim, and Hamayun Khan. "A Robust Hybrid Machine Learning based Implications and Preventions of Social Media Blackmailing and Cyber bullying: A Systematic Approach." The Asian Bulletin of Big Data Management 5, no. 1 (2025): 30-42.

[3] Aziz, Rimsha, Aneela Mehmood, Asma Tariq, Fawad Nasim, Umar Farooq, Syed Asad Ali Naqvi, and Hamayun Khan. "Critical Evaluation of Data Privacy and Security Threats: An Intelligent Federated Learning-based Intrusion Detection System Poisoning Attack and Defense for Cyber-Physical Systems its Issues and Challenges Related to Privacy and Security in IoT." The Asian Bulletin of Big Data Management 5, no. 1 (2025): 73-84.

[4] Noor, Hajira, Jawad Ahmad, Ammar Haider, Fawad Nasim, and Arfan Jaffar. "A machine learning sentiment analysis approach on news headlines to evaluate the performance of the Pakistani government." Journal of Computing & Biomedical Informatics 7, no. 02 (2024).

[6] M. Muller, S. Kornblith, and Y. Elazar, "Explainable fake news detection with neural language models," Proc. ACL, 2021. https://aclanthology.org/2021.acl-long.45

[7] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with social context," Big Data, vol. 8, no. 3, 2020. https://doi.org/10.1089/big.2020.0062

[8] J. Thorne et al., "Evidence-based fact checking with transformers," Proc. EMNLP, 2021. https://aclanthology.org/2021.emnlp-main.609

[9] W. Y. Wang and A. McKeown, "Leveraging contextualized representations for fake news detection," Proc. NAACL-HLT, 2020. https://doi.org/10.18653/v1/2020.naacl-main.177

[10] X. Zhou, A. Mulay, and R. Zafarani, "ReCOVery: A multimodal repository for COVID-19 news credibility," Proc. CIKM, 2020. https://doi.org/10.1145/3340531.3412880

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT, 2019. https://doi.org/10.18653/v1/N19-1423

[12] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT: A distilled version of BERT," Proc. ACL Workshop on Energy Efficient Deep Learning, 2020. https://arxiv.org/abs/1910.01108

[13] A. Vaswani et al., "Attention is all you need," Advances in Neural Information Processing Systems (NeurIPS), 2017. https://arxiv.org/abs/1706.03762

[14] T. Wolf et al., "Transformers: State-of-the-art NLP," Proc. EMNLP Demos, 2020. https://doi.org/10.18653/v1/2020.emnlp-demos.6

Online ISSN 3007-3197
Print ISSN 3007-3189

[15] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011. http://jmlr.org/papers/v12/pedregosa11a.html
[16] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," Science, vol. 359, no. 6380, pp. 1146-1151, 2018. https://doi.org/10.1126/science.aap9559
[17] D. Lazer et al., "The science of fake news," Science, vol. 359, no. 6380, pp. 1094-1096, 2018. https://doi.org/10.1126/science.aao2998
[18] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," Information Sciences, vol. 497, pp. 38-55, 2019. https://doi.org/10.1016/j.ins.2020.04.035
[19] K. Shu, S. Wang, and H. Liu, "User credibility modeling for fake news detection," IEEE Transactions on Big Data, vol. 8, no. 2, pp. 401-413, 2022. https://doi.org/10.1109/TBDATA.2019.2911794
[20] A. Zubiaga et al., "Detection and resolution of rumours in social media," ACM Computing Surveys, vol. 51, no. 2, 2018. https://doi.org/10.1145/3161603
[25] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson, 2021.
[26] Y. Zhang, J. Xu, P. Zhou, and D. Song, "Detecting fake news with attention-based mechanism," IEEE Access, vol. 8, pp. 46880-46893, 2020. https://doi.org/10.1109/ACCESS.2020.2980799
[27] Nasim, Fawad, Sohail Masood, Arfan Jaffar, Usman Ahmad, and Muhammad Rashid. "Intelligent Sound-Based Early Fault Detection System for Vehicles." Computer Systems Science & Engineering 46, no. 3 (2023).
[28] Kazmi, Syeda Saliha, Shahbaz Shamshad, Fawad Nasim, Saira Hashim, Nisha Azam, and Bushra Ambar. "Real-Time Vehicle Detection with Advanced Machine Learning Algorithms." Journal of Computing & Biomedical Informatics 7, no. 02 (2024).
[29] G. Louppe, "Understanding random forests: From theory to practice," arXiv preprint, 2014. https://arxiv.org/abs/1407.7502
[30] S. Ruder, "An overview of multi-task learning in deep neural networks," arXiv preprint, 2017. https://arxiv.org/abs/1706.05098
[38] M. 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. https://doi.org/10.1257/jep.31.2.211
[39] S. Harris et al., "Fake news detection revisited: An extensive review of theoretical frameworks, dataset assessments, model constraints, and forward-looking research agendas," Technologies, 2024. https://doi.org/10.3390/technologies12110222