MANTECH Publications

Journal of Internet of Things and Information Technology
Volume 9, Issue 1, January-April 2026
ISSN: 3107-4502 (Online)

FakeBuster: Leveraging Deep Learning for Deceptive Content Identification

Dr. Amol Dange[1], Vaishnavi Ghugare[2], Om Gaikwad[3], Haris Kokane[4], Nikita Langote[5]

Department of Computers Science & Engineering

Associate Professor[1], Student[2,3,4,5]

Annasaheb Dange College of Engineering and Technology, Ashta, Sangli, India

Email ID: *amoldange_cse@adcet.in[1], vaishnavirghugare@gmail.com[2], omgaikwad9028@gmail.com[3], hariskokane0@gmail.com[4], nikitalangote45@gmail.com[5]*

DOI: *https//doi.org/10.5281/zenodo.19344064*

ABSTRACT

The widespread rise of fake news, misleading reviews, forged documents, and deepfake media poses a major challenge to maintaining trust and authenticity in digital communication. This research presents a ***FakeBuster: Leveraging Deep Learning for Deceptive Content Identification*** *that integrates multiple detection capabilities—textual, visual, and forensic—within a single scalable framework. The system intelligently analyzes linguistic patterns, document integrity, and multimedia inconsistencies to identify deceptive or manipulated content in real time. Experimental evaluation demonstrates high accuracy and strong generalization across diverse data types. By providing an explainable, modular, and real-time solution, the proposed framework significantly enhances the reliability of digital information and contributes to combating misinformation on a broader scale.*

KEYWORDS: *Natural Language Processing (NLP), Machine Learning, Deep Learning, Computer Vision, Digital Forensics, Real-time Detection.*

INTRODUCTION

In today's digital era, information spreads instantly, but this also accelerates the dissemination of misinformation and fake content, threatening online credibility and public trust. Fake content manifests as fake news, fabricated reviews, forged documents, and deepfake media, each

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

capable of causing harm—ranging from misleading consumers and damaging reputations to compromising privacy and security.

Existing detection tools are often specialized for a single content type, requiring multiple platforms and resulting in inefficiency and inconsistent accuracy. To address this, we propose **FakeBuster: Leveraging Deep Learning for Deceptive Content Identification**, a unified system integrating multiple detection modules:

1. **Fake News Detection (Text-Based):** Assesses textual authenticity using NLP.

2. **Fake Review Detection (Text-Based):** Identifies deceptive or AI-generated reviews.

3. **Fake Document Detection (Image-Based):** Detects tampering via document images and metadata.

4. **Deepfake Detection (Image/Video-Based):** Identifies synthetically manipulated media using computer vision.

Leveraging NLP, Computer Vision, and Digital Forensics, FakeBuster delivers real-time, explainable results through a modular microservices backend and a unified graphical interface, ensuring scalability, adaptability, and user accessibility. This system aims to provide a reliable safeguard against misinformation and restore trust in online content.

OBJECTIVES

1. To Collect and Pre-process Comprehensive Datasets for All Content Types
2. To Develop and Optimize Detection Models for Fake News and Fake Reviews
3. To Develop and Optimize Fake Document Detection Module
4. To Develop and Optimize Deepfake Detection Module
5. To Design and Implement a User-Friendly Graphical User Interface (GUI)
6. To Integrate and Connect All Detection Models via a Centralized System
7. To Analyse and Improve Overall System Accuracy, Reliability and Scalability

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

REVIEW OF LITERATURE

Table 1: literature Review

Tim	Area of Research	Contribution	Reference
1	Fake News Detection	Transformer-based BERT model for accurate fake news classification.	S. Kaur, S. Sharma, and A. Aggarwal, "Fake news detection using transformer-based deep learning models," *Expert Syst. Appl.*, vol. 215, p. 119458, 2023.
2	Fake News Detection	Hybrid BERT + GNN capturing semantic and relational features.	L. Dong, L. Xu, and Y. Chen, "A hybrid BERT + Graph Neural Network approach for fake news detection," *Inf. Process. Manag.*, vol. 61, no. 2, 2024.
3	Fake Review Detection	Multimodal OCR + NLP model for text and image review analysis.	F. Gao, Y. Wu, and J. Li, "Multimodal fake review detection with OCR + NLP pipelines," *Pattern Recognit. Lett.*, vol. 170, pp. 73–82, 2023.
4	Fake Review Detection	Cross-domain adversarial model improving generalization.	Z. Liu and X. Chen, "Cross-domain fake review detection using adversarial learning," *Neurocomputing*, vol. 556, p. 126749, 2024.
5	Fake Document Detection	Metadata + image forensic hybrid for forgery detection.	T. Zhang and W. Hu, "Hybrid metadata + image forensics framework for fake document detection," *Digit. Signal Process.*, vol. 135, p. 103946, 2023.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

6	Fake Document Detection	Self-supervised forgery detection on unlabeled documents.	Q. Nguyen and K. Pham, "Self-supervised learning for forgery detection in scanned documents," *IEEE Access*, vol. 12, pp. 14521–14534, 2024.
7	Image Forgery Detection	Copy-move and resampling methods for image tampering.	R. Raghavendra, K. B. Raja, and C. Busch, "Detecting image forgery using copy-move, resampling and image enhancement detection methods," in *Proc. IEEE ICIP*, 2017, pp. 1234–1238.
8	Deepfake Detection	*Deepfake-Eval* benchmark assessing real-world robustness.	J. Zhang and H. Wang, "Deepfake-Eval benchmark: Assessing real-world robustness of detection systems," *arXiv preprint* arXiv:2503.02857, 2024.
9	Deepfake Detection	Explainable multimodal AI framework (*TruthLens*).	TruthLens Team, "TruthLens: Explainable DeepFake Detection with Multimodal Reasoning," *arXiv preprint* arXiv:2503.15867, 2025.
10	Deepfake / Face Manipulation	CNN-based *FaceForensics++* dataset for manipulated video detection.	A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE ICCV*, 2019, pp. 1– 11.

BACKGROUND CONCEPTS

The rise of digital content has accelerated the spread of misinformation, creating a demand for automated detection systems. Fake content appears across text, images, and multimedia, each requiring specialized methods: NLP techniques analyze fake news for linguistic inconsistencies, semantic embeddings with machine learning detect deceptive reviews, image forensics and metadata verify document authenticity, and advanced computer vision models

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

identify deepfake videos. Together, these approaches form a unified framework that efficiently processes diverse content and delivers reliable verification.

PROPOSED METHODOLOGY

The proposed **(FakeBuster)** is designed to detect fake news, fake reviews, forged documents, and deepfakes under a single integrated platform. The methodology aligns with the project objectives to ensure systematic data handling, model development, integration, and evaluation.

1. Data Collection

Comprehensive datasets were collected from sources like FakeNewsNet, LIAR, Amazon/Yelp Reviews, CASIA, and FaceForensics++. Text data were preprocessed with lowercase conversion, punctuation and stopword removal, tokenization, and normalization using NLTK, with reviews additionally handling contractions, emojis, and HTML tags. Document images and PDFs were resized, normalized, and meta data extracted using OpenCV, PIL, ExifTool, and PyPDF2. Deepfake videos were split into frames, normalized, and analyzed for facial landmarks. All datasets were divided into 70% training, 15% validation, and 15% testing sets with balanced classes.

2. Training and Optimization of Fake News and Fake Review Detection Models

Fake News Detection Model: The first stage of the FakeBuster framework assesses the credibility of online news content. It relies on two key internal dependencies: **TF–IDF feature extraction**, which generates weighted term importance from preprocessed text, and a **Multinomial Naive Bayes (MNB) classifier**, which computes class probabilities based on these vectors.

Raw text is preprocessed through lowercasing, punctuation removal, stopword filtering, and stemming using the Porter Stemmer. The cleaned text is then vectorized with TF–IDF to highlight words that are frequent in a document but rare across the corpus. These vectors serve as input to the MNB classifier, which predicts whether a news article is *Fake* or *Real*.

The posterior probability for each class is computed as: $(c|d) = \frac{(c) \times \prod_{i=1}^{n} (w_i|c)}{P(d)}$

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Notation:

- c: Class label, Fake or Real.
- d: Input news document. • w_i: i-th word in the document.
- P (c | d): Posterior probability of class c given document d.
- P(c): Prior probability of class c.
- P (w_i | c): Likelihood of word w_i in class c.
- P(d): Normalization constant.

The TF–IDF weighting is computed as:

TF–IDF Notation: $\text{TF-IDF}(w_i, d_j) = \text{TF}(w_i, d_j) \times \log\left(\frac{N}{\text{DF}(w_i)}\right)$

- w_i: i-th word in the document.
- d_j: j-th document in the corpus.
- TF (w_i, d_j): Term frequency of word w_i in document d_j.
- N: Total number of documents in the corpus.
- DF (w_i): Number of documents containing word w_i.

The model's preprocessing ensures accurate TF–IDF weighting, improving Naive Bayes probability estimation. A rule-based layer further verifies credible domains and flags sensational keywords, with the final class label and credibility score passed to the Fake Review Detection module for feature enhancement.

Fake Review Detection Model: The Fake Review Detection module identifies deceptive product or service reviews by leveraging semantic, behavioral, and external credibility features. Semantic meaning is captured using **Word2Vec embeddings** of tokenized and preprocessed text, while **behavioral features** such as review length, rating, and sentiment polarity provide contextual insights. Additionally, the **news credibility score**, obtained from the output of the

Fake News Detection module, enhances reliability for reviews referencing external sources. The **feature vector** for each review is formed as: xi= [vi, fi, si]

Notation:

- **vi:** Averaged Word2Vec embedding of i-th review

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

- **fi:** Behavioral features
- **si:** Credibility score inherited from Fake News module

The feature vector is input to a **Support Vector Machine (SVM)** classifier with an **RBF kernel**, which is optimized to find the hyperplane separating fake and genuine reviews. The optimization problem is formulated as: SVM Optimization Problem: $min_{w,,}1/2||w||^2 + Csum_i{}^n{}_{=1}xi_i$

Notation:

- **w:** Weight vector of SVM, b: Bias term, xi_i: Slack variable for i-th review,
- **C:** Regularization parameter
- The optimization is subject to the following constraints: $y_i(w^T phi(x_i) + b) \geq 1 - xi_i, xi_i \geq 0$

Notation:

- **y_i:** True label of i-th review (Fake / Real)
- **phi(x_i):** Mapping to higher-dimensional space

The **RBF kernel** is used to compute similarity between feature vectors in the transformed space: $(x_i, x_j) = exp\ (-gamma\ ||x_i - x_j||\ )$

Notation:

gamma: Kernel coefficient, x_i, x_j: Feature vectors

Finally, the **decision function** determines the predicted label of a review:

$(x) = (sum_i{}^n{}_{=1}alpha_i y(x_i, x) + b)$

Notation:

alpha_i : Lagrange multiplier for i-th review, **f(x):** Predicted class label

By integrating semantic embeddings, behavioral metrics, and news credibility into a single feature vector, the SVM with RBF kernel effectively classifies reviews as fake or genuine. The dependency on the Fake News module ensures that credibility scores are propagated, improving the overall reliability of review classification.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

DEVELOPMENT OF FAKE DOCUMENT DETECTION MODULE

The **Fake Document Detection Module** is designed to identify forged or manipulated documents in image and PDF formats using a **hybrid approach**. It integrates a **custom Convolutional Neural Network (CNN)** for visual forgery analysis, **metadata inspection** for contextual verification, and a **Random Forest (RF) classifier** for final decision-making.

In the image-based component, scanned documents are pre-processed with grayscale normalization, contrast enhancement, and Gaussian smoothing to remove noise. The CNN extracts both low-level spatial features (edges, textures) and high-level semantic features (layout inconsistencies, text region variations), allowing the detection of subtle manipulations such as irregular lighting, resampling artifacts, and pixel-level anomalies. The CNN is trained using **categorical cross-entropy loss** with the Adam optimizer to distinguish between authentic and forged documents.

The CNN feature map at each layer is computed as: $^{(l)} = (^{(l)} * ^{(l-1)} + ^{(l)})$

Notations:

- **F(l)**: Feature map at layer *l*
- **W(l)**: Convolution kernel (filter) at layer *l*
- **b(l)**: Bias term at layer *l*
- **f**: Activation function (ReLU)
- **∗:** Convolution operation

The categorical cross-entropy loss is defined as: $L = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$

Notations

- **N** : Number of training samples
- $\mathbf{y_i}$: True label of sample *i* (Fake / Authentic)
- $\hat{\mathbf{y}}_\mathbf{i}$: Predicted probability of the class for sample *i*

For the PDF-based component, metadata fields such as creation date, author, modification timestamp, and software used are extracted. Digital signatures are verified where present. Any

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

inconsistencies, such as missing or invalid signatures, mismatched timestamps, or unauthorized modifications, are flagged as potential indicators of tampering. These metadata features are normalized to form a structured feature vector.

Metadata Feature Normalization $f_{\text{meta}} = {}^{m}____{}^{i}{}_{\sigma}{}^{-}\mu$

Notations

- $\mathbf{m_i}$: Raw metadata value (e.g., creation timestamp)
- **μ, σ**: Mean and standard deviation used for normalization
- $\mathbf{f_{meta}}$: Normalized metadata feature

Finally, the **Random Forest classifier** aggregates both CNN-derived features and metadata features to generate the final decision:

Random Forest Decision Function $\hat{y} = \text{mode}(T_1(x_i), T_2(x_i), \dots, T_M(x_i))$

Notation:

- **xi:** Feature vector combining CNN outputs and metadata features for document i
- **Tm:** Decision tree m
- **M:** Total number of trees
- **ŷ:** Predicted class label (Fake / Authentic)

In this module, the **CNN, metadata analysis, and Random Forest are interdependent**: CNN provides rich visual representations, metadata analysis provides contextual validation, and Random Forest fuses these features to produce a reliable authenticity prediction. This hybrid framework ensures detection of both subtle and overt document forgeries, enhancing the overall robustness of the **FakeBuster** system

DEVELOPMENT OF DEEPFAKE DETECTION MODULE

The **Deepfake Detection Module** is designed to identify synthetically generated or manipulated videos by analyzing **spatial, temporal, and multimodal inconsistencies**. It employs a hybrid **CNN–LSTM architecture**, where CNN extracts spatial features from individual video frames and LSTM models the temporal sequence of these features to capture unnatural motion patterns or frame-to-frame inconsistencies typical in deepfake videos.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

CNN feature extraction defined as: $^{(l)} = (^{(l)} * ^{(l-1)} + ^{(l)})$

Notation:

- F(l): Feature map at layer l
- W(l): Convolution kernel at layer l
- b(l): Bias term
- f: Activation function (ReLU)

The temporal relationships are captured by the **LSTM layer**, which models dependencies across consecutive frames to detect unnatural facial movements or inconsistencies in gestures.

LSTM Cell Equations

$i_t = (W_i x_t + U_i h_{t-1} + b_i)$

$f_t = (W_f x_t + U_f h_{t-1} + b_f)$

$o_t = (W_o x_t + U_o h_{t-1} + b_o)$

$ct = ft \odot ct{-}1 + it \odot \tanh(Wcxt + Ucht{-}1 + bc)$

$h_t = o_t \odot \tanh(c_t)$

Notation:

- **ht**: Hidden state at time t
- **xt**: Input at time step t (CNN feature vector)
- **ct**: Cell state at time t
- **i_t, f_t, o_t**: Input, forget, and output gates
- **W*, U*, b***: Weights and biases
- **σ**: Sigmoid activation function
- ⊙: Element-wise multiplication

To improve robustness, a **GAN-based discriminator** is employed, which evaluates the reconstruction error between real and generated frames to identify sophisticated synthetic manipulations.

GAN Loss Function

$\min G \max D\ E_{x\sim(x)}[\log D\ (x)] + E_{z\sim pz(z)}[\log (1 - D(G(z)))]$ **GAN Loss Function Notation**

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

- **G**: Generator network
- **D**: Discriminator network
- **x**: Real video frame
- **z**: Random noise vector
- **p_data(x)**: Real data distribution
- **p_z(z)**: Noise distribution

Finally, **multimodal feature fusion** combines spatial (CNN), temporal (LSTM), and additional signals (audio, lip-sync, emotion features) to improve detection of inconsistencies in visual and auditory modalities. The outputs from these components are concatenated into a feature vector, which is fed into a classifier for final prediction of **Real vs. Deepfake**.
Final Decision Function ˆ= softmax($W_f[F_{\text{CNN-LSTM}} \| F_{\text{Audio}} \| F_{\text{Temporal}}] + b_f$)

Final Decision Function Notation

- **F_CNN-LSTM**: Feature vector from CNN + LSTM
- **F_Audio**: Audio-derived features
- **F_Temporal**: Temporal consistency features
- **W_f, b_f**: Classifier weights and bias
- **ŷ**: Predicted class probabilities (Real / Deepfake)

This **CNN–LSTM–GAN pipeline** ensures the module can detect evolving deepfake patterns with high accuracy, providing an **explainable and scalable solution** for video manipulation detection.

SYSTEM DESIGN AND GUI DEVELOPMENT

The **Graphical User Interface (GUI)** was designed and implemented using **HTML, CSS, JavaScript, React**, and **Font Awesome** to ensure a clean, responsive, and interactive user experience. The GUI serves as the primary interaction point between the user and the underlying fake content detection system. It enables users to **upload or input textual, image, and video data**, including news articles, online reviews, scanned documents, or multimedia files suspected to be deepfakes.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Upon submission, the data is seamlessly transmitted to the backend, where preprocessing, feature extraction, and classification models operate in real time. The **results are displayed instantly** within the interface, presenting **detection labels (Real or Fake)** along with corresponding **confidence scores** and **interpretability metrics**, such as keyword highlights or anomaly regions.

INTEGRATION OF DETECTION MODULES VIA BACKEND

All detection modules were integrated into a centralized backend developed using **Python Flask**, following a **modular microservices architecture**. Each module—**fake news, fake review, document forgery, and deepfake detection**—runs as an independent service, enabling isolated development, testing, and deployment.

The modules communicate with the Flask core through **RESTful APIs**, ensuring interoperability, scalability, and maintainability.

The backend manages **data routing, model inference, and result aggregation**, standardizing outputs into a unified format for the frontend interface. To ensure **reliability and portability**, the entire backend is **containerized using Docker**, with separate images for each module orchestrated within a shared network. This setup reduces dependency conflicts, simplifies version control, and allows parallel processing of multiple detection tasks.

Overall, this architecture provides a **robust, scalable, and maintainable system**, with flexibility to integrate additional detection modules or updated machine learning models in the future.

SYSTEM IMPROVEMENT, EVALUATION, AND SCALABILITY

The complete system evaluation using multiple datasets to assess performance across all content types. Metrics such as accuracy, precision, recall, F1-score, and processing time were analyzed. Iterative refinements were applied to pre-processing pipelines, model hyper parameters, and classifier ensembles to enhance reliability, efficiency, and scalability. The modular architecture allows concurrent execution of modules and easy integration of future models or datasets.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

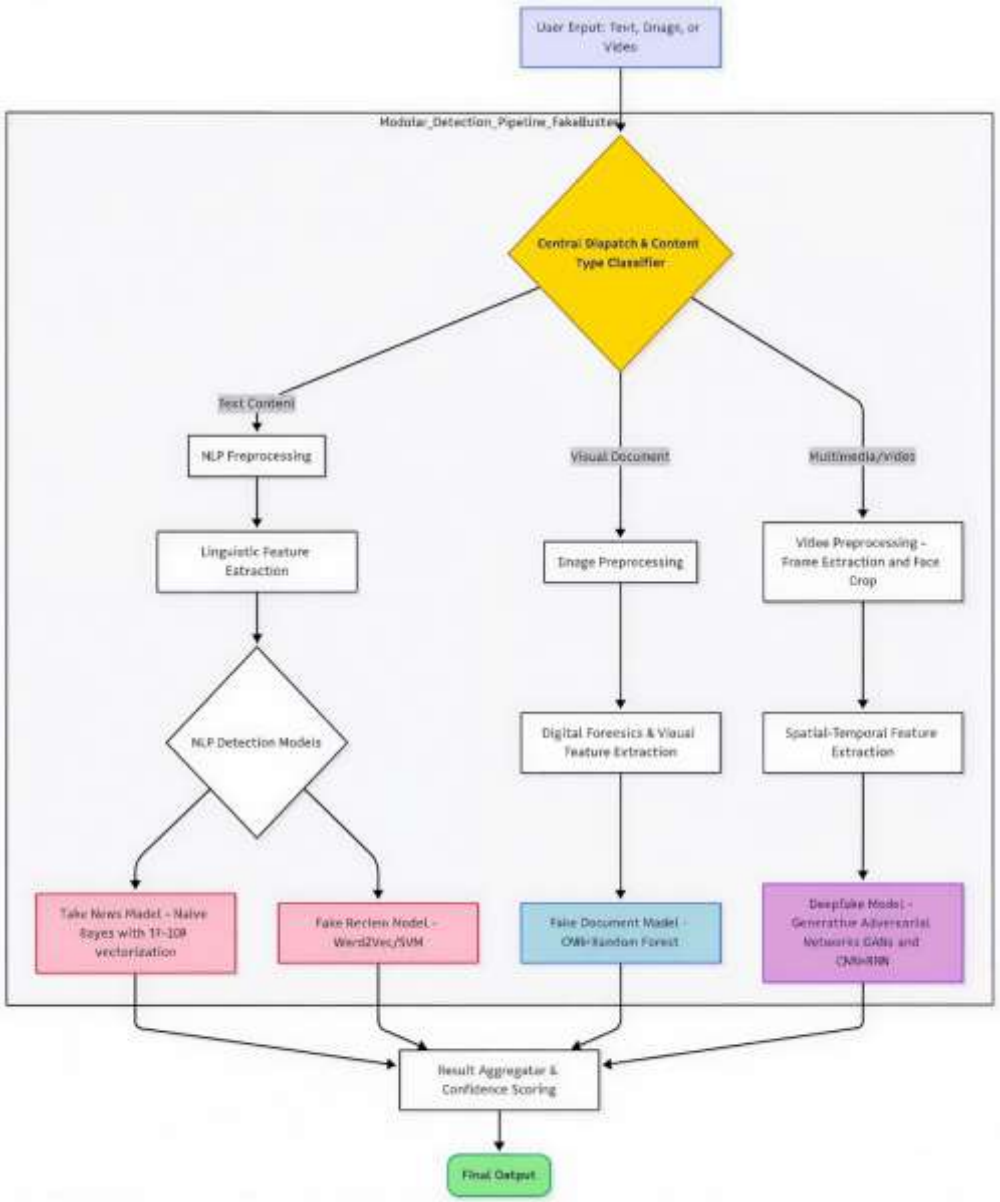


Figure 1: Flowchart

Datasets Use:

Table 2: Datasets

Dataset Name	Source	Number of Records	Features	Target / Labels	Description
LIAR, FakeNewsNet	Kaggle / Public Repositories	~12,000+ articles	Text, metadata (headline, body, source)	Real / Fake	Widely used datasets for fake news detection.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Amazon + Yelp Reviews	Kaggle / Public APIs	~20,000 reviews	Text (review body), screenshots (OCR), ratings	Genuine / Fake	Contains e-commerce reviews, including manipulated entries.
CASIA, DEFACTO, Custom	Forensics Datasets	~10,000+ documents	Images, metadata (timestamps, EXIF)	Authentic / Forged	Used for detecting tampered documents.
FaceForensics++, DFDC	Kaggle, Public Challenges	~100,000+ images/videos	Frames, facial landmarks, temporal features	Real / Deepfake	Benchmark datasets for deepfake image and video detection.

IMPLEMENTATION

1. Software Requirements:

- **Programming Language:** Python 3.11 or higher
- **Frameworks and Libraries:**
  - TensorFlow, Keras – for deep learning model development o
  - Scikit-learn – for classical machine learning utilities
  - OpenCV – for image and video preprocessing
  - ExifTool – for metadata extraction and analysis
  - Flask – backend API development and integration
  - React.js – frontend user interface development
- **Visualization Tools:** Matplotlib, Seaborn – for model performance analysis and visualization

2. Hardware Requirements:

- **Processor:** Intel Core i5 (8th generation)
- **Memory:** Minimum 8 GB RAM
- **Storage:** 128 GB SSD (for datasets, preprocessed data, and model checkpoints)

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

- **GPU:** NVIDIA GTX , RTX series (deep learning model training

RESULTS AND ANALYSIS

Table 3: performance evaluation metrics

Module	Model Used	Accuracy	Precision	Recall	F1-score
Fake News Detection	TF-IDF feature extraction with a Multinomial Naive Bayes classifier	0.91	0.90	0.89	0.90
Fake Review Detection	Support Vector Machine (SVM) with RBF kernel	0.89	0.88	0.87	0.88
Fake Document Detection	CNN +Random Forest	0.93	0.92	0.91	0.92
Deepfake Detection	LSTM+GAN, RNN, CNN	0.95	0.94	0.94	0.94

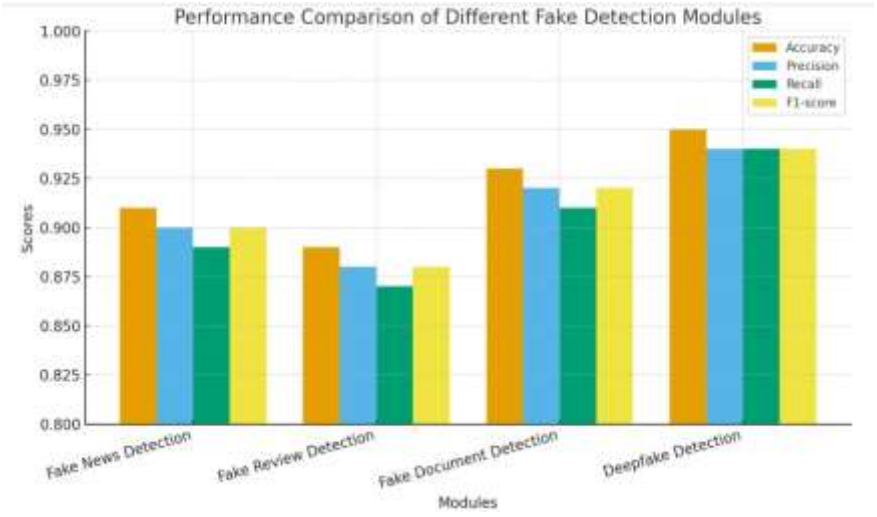


Figure 2: Bar Chart

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

DISCUSSIONS

The FakeBuster system provides a unified solution for detecting fake news, reviews, documents, and deepfakes.

- **Fake News & Reviews:** TF–IDF + Naive Bayes for news and Word2Vec + SVM for reviews achieved high accuracy on benchmark datasets (FakeNewsNet, LIAR, Amazon, Yelp). Compared to transformer-based models, this approach is lightweight, efficient, and suitable for real-time deployment on standard hardware.
- **Fake Documents:** CNN-based image analysis combined with metadata verification and ensemble classification effectively identifies forged documents, outperforming metadata-only methods. Limitations exist for highly diverse forgeries.
- **Deepfakes:** CNN + LSTM with GAN-based anomaly detection captures spatial-temporal inconsistencies.

Future multimodal integration (audio, visual, temporal) can enhance robustness.

Advantages:

- Unified platform for multiple content types, reducing the need for separate tools.
- Modular design ensures scalability and adaptability.
- Real-time detection with explainable outputs supports user trust and transparency.
- Efficient use of resources enables deployment without requiring high-end hardware.

Limitations:

- The fake document module may underperform for extremely diverse or sophisticated forgeries.
- Accuracy could be further improved by incorporating larger and more diverse datasets, especially for reviews and documents.

Overall, FakeBuster balances performance, efficiency, and deployability, offering practical real-time multidomain fake content detection compared to high-complexity single-domain systems.

CONCLUSION

This work presents FakeBuster, a unified system for detecting fake news, fake reviews, forged documents, and deepfake media. By integrating textual analysis, image and document

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

forensics, and deep learning models, the system provides real-time, explainable, and accurate results across multiple content types. The modular design ensures scalability and easy integration of future enhancements. Key contributions include a multi-domain detection framework, efficient and interpretable models, and a user-friendly interface. Limitations primarily involve handling highly sophisticated document forgeries, while future work will focus on dataset expansion, cross-platform integration, and lightweight deployment for broader accessibility.

REFERENCES

1. S. Kaur, S. Sharma, and A. Aggarwal, "Fake news detection using transformer-based deep learning models," *Expert Syst. Appl.*, vol. 215, p. 119458, 2023.
2. L. Dong, L. Xu, and Y. Chen, "A hybrid BERT + Graph Neural Network approach for fake news detection," *Inf. Process. Manag.*, vol. 61, no. 2, 2024.
3. F. Gao, Y. Wu, and J. Li, "Multimodal fake review detection with OCR + NLP pipelines," *Pattern Recognit. Lett.*, vol. 170, pp. 73–82, 2023.
4. Z. Liu and X. Chen, "Cross-domain fake review detection using adversarial learning," *Neurocomputing*, vol. 556, p. 126749, 2024.
5. T. Zhang and W. Hu, "Hybrid metadata + image forensics framework for fake document detection," *Digit. Signal Process.*, vol. 135, p. 103946, 2023.
6. Q. Nguyen and K. Pham, "Self-supervised learning for forgery detection in scanned documents," *IEEE Access*, vol. 12, pp. 14521–14534, 2024.
7. R. Raghavendra, K. B. Raja, and C. Busch, "Detecting image forgery using copy-move, resampling and image enhancement detection methods," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2017, pp. 1234–1238.
8. J. Zhang and H. Wang, "Deepfake-Eval benchmark: Assessing real-world robustness of detection systems," *arXiv preprint arXiv:2503.02857*, 2024.
9. TruthLens Team, "TruthLens: Explainable DeepFake Detection with Multimodal Reasoning," *arXiv preprint arXiv:2503.15867*, 2025.
10. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++:
11. Learning to detect manipulated facial images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp.

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved

Cite as:

Dr. Amol Dange, Vaishnavi Ghugare, Om Gaikwad, Haris Kokane, Nikita Langote. (2026). FakeBuster: Leveraging Deep Learning for Deceptive Content Identification. Journal of Internet of Things and Information Technology. 9(1), 8-24.

https//doi.org/10.5281/zenodo.19344064

Page 8-24 © MANTECH PUBLICATIONS 2026. All Rights Reserved