TruthLens: AI-Powered Fake News and Misinformation Detection Using Multimodal Analysis

SADIYA MAHEEN SIDDIQUI
Computer Science and Engineering
University College of Engineering - Osmania University
Hyderabad, India
adibasadiya9502@gmail.com

***Abstract*—The spread of misinformation and fake news poses a significant challenge in today's digital landscape. This research presents TruthLens, an AI-powered framework integrating Natural Language Processing (NLP), Computer Vision (CV), and Fact-Checking APIs to identify and mitigate misinformation. Our system leverages machine learning models for textual analysis, deep learning-based image/video forensics, and web scraping techniques for real-time verification. The credibility scores are evaluated using TF-IDF with LinearSVC, BERT, RoBERTa, CNNs for manipulated media, and Google Fact-Check API, achieving a robust, multi-modal detection pipeline.**

***Index Terms*—Misinformation Detection, Fake News, Artificial Intelligence (AI), Natural Language Processing (NLP), Computer Vision (CV), Fact-Checking APIs, Machine Learning, Deep Learning, Media Manipulation Detection, Multi-Modal Detection**

I. Introduction

The rapid dissemination of false information through social media and news platforms necessitates advanced detection systems. Existing solutions often focus solely on text-based analysis, neglecting **visual misinformation** (deepfakes, altered images) and **cross-referencing web sources**. TruthLens is designed to address this gap using a **multi-modal approach**:

1) **Text Analysis** – Classifying news articles and social media content using machine learning.
2) **Image & Video Forensics** – Detecting manipulated media with deep learning.
3) **Fact-Checking & Web Verification** – Extracting credibility from sources and scoring misinformation.

This paper details our **methodology, architecture, implementation, and performance evaluation** of TruthLens across these domains.

II. Methodology

A. Text-Based Fake News Detection

Detecting misinformation in textual content requires a multi-layered approach that combines machine learning, natural language processing, fact-checking, and web verification. Our method integrates four key components to enhance the accuracy and reliability of fake news detection:

1) **ML Prediction Model** – A supervised learning approach leveraging TF-IDF vectorization and transformer-based embeddings for classification.
2) **Fact-Check API Integration** – Using Google's Fact-Check API to validate claims against trusted sources.
3) **Web Scraping & Verification** – Collecting related articles from search results and assessing content similarity.
4) **Sentiment Analysis** – Evaluating neutrality to further refine credibility scoring.

These components work together to produce a **weighted credibility score**, which determines the likelihood of a given news article being misinformation.

*1) **ML Prediction Model**:* To classify textual misinformation, a machine learning model was trained on the **Fake-True News Classification Kaggle Dataset**. The approach combined traditional and deep learning techniques to enhance accuracy and interpretability:

- **TF-IDF vectorization with LinearSVC** achieved **1˜00% accuracy** on the test set, significantly surpassing the **Na¨ıve Bayes classifier (94%)**.
- **BERT and RoBERTa** were utilized for deep contextual understanding of text, improving classification performance on complex or nuanced statements.

*a) **Text Analysis Workflow**:* The classification pipeline consists of three key stages:

1) **Preprocessing**
   - Tokenization
   - Stopword Removal
   - Lemmatization
2) **Feature Extraction**
   - **TF-IDF representation** for lightweight, explainable classification

- **Transformer embeddings (BERT/RoBERTa)** for capturing deep semantic context

3) **Classification**
   - **LinearSVC** for computational efficiency and interpretability
   - **BERT/RoBERTa** for robust handling of complex linguistic structures

*2) **Fact-Check API Integration**:* To improve the reliability of predictions, the **Google Fact-Check API**was integrated which validates claims against authoritative sources. This ensures that factual statements are supported by reputable news organizations and reduces false positives in misinformation detection.

*3) **Web Scraping & Fact Verification**:* To further enhance credibility assessment, a **real-time web scraping module** was implemented using **BeautifulSoup and Selenium**. This module collects related articles and analyzes their content for alignment with the input text.

*a) **Web Verification Process**:*

1) **Search Query Generation** – Extracts key terms from the given text.
2) **Google Search Scraping** – Collects the top 5 related news articles.
3) **Content Similarity Analysis** – Uses **TF-IDF cosine similarity** to measure how closely these articles align with the input.

The results from this module contribute to the overall credibility score, ensuring that information backed by multiple sources is ranked higher in reliability.

*4) **Sentiment Analysis & Credibility Score Calculation**:* Sentiment analysis is used to assess the neutrality of a given text, as highly emotional or biased language is often an indicator of misinformation. The **TextBlob** library was used to compute sentiment scores. A neutrality-based approach ensures that news with a balanced tone receives a higher credibility score, whereas overly sensationalized content is flagged as potentially misleading.

*5) **Final Credibility Score Calculation**:* A **weighted credibility score** is assigned based on multiple factors:

- **ML Prediction Confidence (30%)** – Confidence score from the machine learning classifier.
- **Fact-Check API Score (25%)** – Degree of validation from authoritative sources.
- **Sentiment Analysis Score (15%)** – A neutrality-based score derived from sentiment analysis.
- **Web Verification Score (30%)** – Content similarity with trustworthy sources.

By combining these components, a robust, multi-layered approach to **text-based fake news detection** was created ensuring high accuracy and reliability in identifying misinformation.

B. Image & Video Misinformation Detection

Visual misinformation, including **altered images, GAN-generated deepfakes, and misleading visuals**, is a growing challenge. The **CNNs and vision transformers (ViTs)** were employed for classification.

*a) **Image-Based Detection**:*

- **ViT (Vision Transformer)**: Trained on a dataset of real vs. manipulated images.
- **CNN for Fake Detection**: Using OpenCV and PyTorch, a CNN was fine-tuned on deepfake datasets.
- **Error Level Analysis (ELA)**: Extracts inconsistencies in pixel values, revealing tampered regions.

*b) **Video-Based Deepfake Detection**:* The video frames were analyzed using:

1) **Frame Extraction & Processing**: OpenCV extracts frames at 5 fps.
2) **Deepfake Detection Model**: A pre-trained **Xception model** detects manipulated faces.
3) **Optical Flow Analysis**: Identifies unnatural movement patterns.

III. Implementation

A. Backend & API Integration

The system is implemented using **Flask**, serving RESTful APIs for **text, image, and video analysis**.

Key technologies:

- **Database**: MySQL for storing news credibility scores.
- **ML Models**: Deployed using TensorFlow and PyTorch.
- **LLM Integration**: Cohere AI models generate **explainable reports** on misinformation cases.

B. Frontend Integration

TruthLens is built with **JavaScript** for an interactive interface, allowing users to:

- **Submit news articles** for credibility analysis.
- **Upload images/videos** for forensic evaluation.
- **View credibility scores** with detailed breakdowns.

IV. Results & Performance Evaluation

A. Text-Based Model Performance

TABLE I
Performance Comparison of Fake News Detection Models

Model	Accuracy	Precision	Recall	F1-Score
TF-IDF + LinearSVC	99.8%	98.5%	99.2%	98.8%
Na¨ive Bayes	94.3%	92.1%	93.5%	92.8%

As shown in Table I, the TF-IDF + LinearSVC model achieved the highest accuracy.

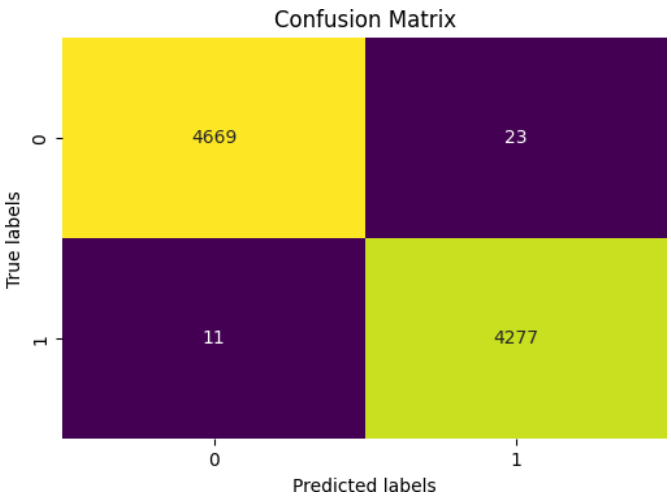

Fig. 1. Confusion matrix for the text-based fake news detection model.

As shown in Figure 1, the TF-IDF + LinearSVC model achieved excellent classification performance with minimal false positives and false negatives.

B. Image-Based Model Performance

TABLE II
PERFORMANCE COMPARISON OF IMAGE-BASED FAKE DETECTION MODELS

Model	Accuracy	Precision	Recall
ViT for Fake Image Detection	97.5%	96.8%	97.2%

As shown in Table II, the ViT for Fake Image Detection model achieved the highest accuracy among the image-based fake detection models, outperforming the CNN + ELA model in terms of precision, recall, and overall detection capability.

TABLE III
CLASSIFICATION REPORT FOR IMAGE-BASED FAKE DETECTION MODEL

Class	Precision	Recall	F1-Score	Support
Artificial	0.9897	0.9347	0.9614	1333
Deepfake	0.9409	0.9910	0.9653	1333
Real	0.9970	0.9993	0.9981	1334

As shown in Table III, the image-based fake detection model achieved a high accuracy of 97.5%. The ”Real” class exhibited the highest precision and recall, while the ”Deepfake” class had a high recall, ensuring robust detection of manipulated content.

Overall, the model demonstrates strong performance across all categories, indicating its effectiveness in distinguishing between artificial, deepfake, and real images.

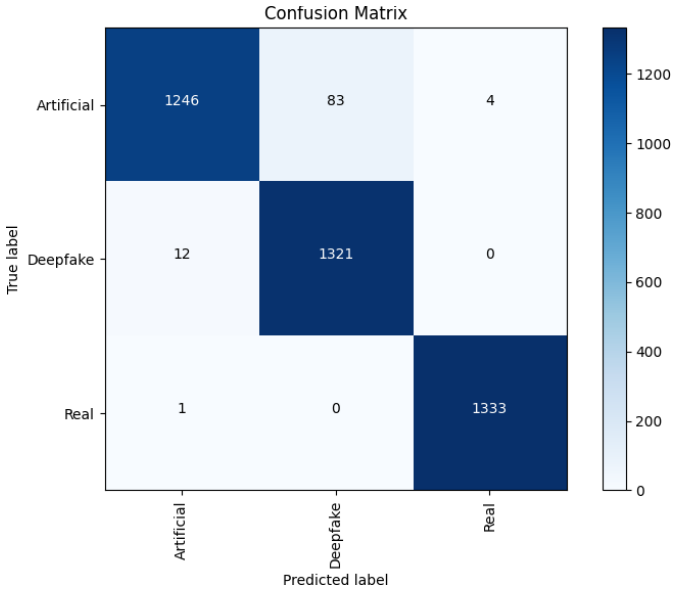

Fig. 2. Confusion matrix for the image-based fake news detection model.

As shown in Figure 2, the TF-IDF + LinearSVC model achieved high classification accuracy, effectively distinguishing between Artificial, Deepfake, and Real categories. The confusion matrix highlights minimal false positives and false negatives, with the model demonstrating strong performance in accurately classifying deepfake content.

C. Video-Based Model Performance

TABLE IV
PERFORMANCE OF VIDEO-BASED DEEPFAKE DETECTION MODEL

Model	Accuracy	Precision	Recall
Xception Deepfake Model	98.3%	97.8%	98.1%

As shown in Table IV, the Xception Deepfake Model achieved an impressive accuracy of 98.3%, with high precision and recall scores. This highlights the model’s effectiveness in detecting deepfake videos with minimal misclassification, making it a reliable choice for video-based forgery detection.

D. Conclusion & Future Work

TruthLens presents a **multi-modal AI-driven approach** to combat misinformation across text, images, and videos. By integrating **ML models, deep learning, fact-checking APIs, and web verification**, it achieves high accuracy in misinformation detection.

Some of our future enhancements include:

- **Multilingual Support**: Expanding to detect misinformation in multiple languages.

- **Social Media Integration**: Deploying browser extensions for real-time verification.
- **Improved GAN Detection**: Enhancing deepfake detection with adversarial training.

TruthLens represents a **scalable, robust, and explainable** solution for the misinformation crisis, with potential real-world applications in journalism, social media moderation, and fact-checking organizations.

ACKNOWLEDGMENT

I would like to express my deepest gratitude to the **University College of Engineering, Osmania University**. I acknowledge the contributions of various open-source libraries and APIs, which were instrumental in implementing and evaluating this study. Finally, I am grateful to my peers, mentors, and everyone who supported me directly or indirectly in completing this research.

REFERENCES

1) **Scikit-learn:** Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E´ . (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.
2) **Hugging Face ViT Model:** PrithivMLmods. *AI-vs-Deepfake-vs-Real*. Available at: PrithivMLmods
3) **Google Fact Check API:** Google. *Fact Check Tools API*. Available at: Fact Check Tools API
4) **Cohere API for Text Analysis:** Cohere. *Large Language Model API for Fact Checking and Summarization*. Available at: Cohere
5) **Flask Framework:** Ronacher, A. (2010). *Flask: Web Development Framework*. Available at: Flask: Web Development Framework
6) **BeautifulSoup for Web Scraping:** Richardson, L. (2007). *Beautiful Soup Documentation*. Available at: Beautiful Soup
7) **EfficientNet for Video-Based Deepfake Detection:** Tan, M., & Le, Q. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. Proceedings of the 36th International Conference on Machine Learning (ICML).
8) **TensorFlow for Deep Learning Models:** Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). *TensorFlow: A System for Large-Scale Machine Learning*. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265-283.
9) **TextBlob for Sentiment Analysis:** Loria, S. (2018). *TextBlob: Simplified Text Processing in Python*. Available at: TextBlob
10) **TQDM for Progress Tracking:** Caswell, T. A., Droettboom, M., Lee, A., & Hunter, J. D. (2021). *TQDM: A Fast, Extensible Progress Bar for Python and CLI Applications*. Available at: TQDM