



Project Proposal

TruthLens: Detecting Fake News Using NLP

Prepared By

Name: **Abdulrahman Alhammadi** ID No.: **251002075**

Submitted in fulfillment of the requirements for the degree Master of Science
in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: **Khaled Eskaf**

Collage of Informatics, Midocean University, Comoros

Email:.....

1. Project Overview

This project aims to develop a lightweight Natural Language Processing (NLP) model that can classify news articles or social media posts as either real or fake. With the rise of misinformation, especially on digital platforms, this tool will help users and organizations quickly assess the credibility of textual content.

2. Project Goal

- **Develop a model to identify** fake news articles using NLP techniques.
- **Create a tool that helps solve** the problem of misinformation by providing a quick credibility check for news content.

3. Approach

- **Data Collection:**
 - Use publicly available datasets like the Fake News Dataset from Kaggle or LIAR dataset.
- **Data Processing:**
 - Clean text (remove stop words, punctuation, etc.)
 - Tokenization and lemmatization
 - Vectorization (TF-IDF or word embeddings)
- **Model/Method Selection:**
 - **Traditional Methods:**
 - TF-IDF
 - Bag of Words (BoW)
 - **Advanced:**
 - Word embeddings (Word2Vec, GloVe)
 - Contextual embeddings (BERT, RoBERTa, DistilBERT, etc.)

4. Project Steps and Timeline

- **Project Plan** divided into 6 **phases** with weekly goals.

		Month	1st Month				2nd Month				3rd Month			
		Week	1	2	3	4	5	6	7	8	9	10	11	12
1	Research & Data Collection (Week 1–2)													
2	Feature Engineering & Baseline Models (Week 3–4)													
3	Advanced NLP Models (Week 5–7)													
4	Model Finalization & Testing (Week 8–9)													
5	Deployment & UI (Week 10–11)													
6	Documentation & Final Report (Week 12)													

• Detailed Activities :

Phase 1: Research & Data Collection (Week 1–2)

- ♦ **Goal:** Lay the foundation, define scope, collect and understand data
- ♦ **Activities:**
 - Define project scope and objectives
 - Literature review (NLP + fake news detection methods)
 - Explore public datasets (Kaggle, LIAR, ISOT, etc.)
 - Choose the main dataset(s)
 - Data cleaning and preprocessing
 - Exploratory Data Analysis (EDA)
- ♦ **Deliverables:** Dataset selected and preprocessed, EDA report

Phase 2: Feature Engineering & Baseline Models (Week 3–4)

- ♦ **Goal: Start modeling with simple approaches and extract basic insights**
- ♦ **Activities:**
 - Implement text preprocessing (tokenization, stopwords, lemmatization)
 - Feature extraction using TF-IDF / BoW
 - Build baseline ML models (Logistic Regression, Naive Bayes, etc.)
 - Evaluate models (accuracy, F1, confusion matrix)
 - Fine-tune preprocessing steps
 - Compare models
- ♦ **Deliverables:** Baseline models, evaluation report

Phase 3: Advanced NLP Models (Week 5–7)

- ♦ **Goal: Move to deep learning and transformer-based models**
- ♦ **Activities:**
 - Word embeddings (Word2Vec / GloVe)

- Build LSTM / BiLSTM model using Keras or PyTorch
- Explore transformer models (BERT, RoBERTa via HuggingFace)
- Fine-tune pre-trained BERT on your dataset
- Evaluate and compare all advanced models
- Perform hyperparameter tuning
- ♦ **Deliverables:** Final model selection, comparative model evaluation

Phase 4: Model Finalization & Testing (Week 8–9)

- ♦ **Goal:** Polish and finalize your best model
- ♦ **Activities:**
 - Perform thorough testing (cross-validation, different test sets)
 - Analyze failure cases (false positives/negatives)
 - Create visualization (ROC curve, word clouds, attention scores)
 - Finalize best model with metrics and validation results
- ♦ **Deliverables:** Final model + performance report

Phase 5: Deployment & UI (Week 10–11)

- ♦ **Goal:** Make the model accessible via a web interface or API
- ♦ **Activities:**
 - Save model (joblib, pickle, or torch.save)
 - Build backend (Flask or FastAPI)
 - Build simple UI with Streamlit or HTML/CSS
 - Integrate model into the UI or API
 - Test end-to-end pipeline
- ♦ **Deliverables:** Fully functional prototype

Phase 6: Documentation & Final Report (Week 12)

- ♦ **Goal:** Summarize everything, prepare for presentation or submission
- ♦ **Activities:**
 - Write final report (methodology, results, discussion)
 - Prepare presentation (slides, demo video)
 - Submit code, notebook, and documentation on GitHub or submission platform
- ♦ **Deliverables:** Final report, presentation slides, GitHub repo or zip

5. Expected Results

- A trained NLP model that can classify news as real or fake with reasonable accuracy.
- (Optional) A simple web app where users can input text and get a credibility score or label.

6. Resources Needed

- **Datasets:** Kaggle Fake News Dataset, LIAR dataset
- **Tools:** Python,
- **Libraries:**
 - pandas, numpy, sklearn, matplotlib, seaborn
 - nltk, spacy, transformers, torch or tensorflow
 - Flask / FastAPI (for deployment)
- **Hardware:** Standard laptop or cloud-based notebook (e.g., Google Colab)

7. Project Assumptions

1. **Labeled Data is Available and Reliable**
We assume that the dataset(s) used contain accurate and correctly labeled samples of "fake" and "real" news articles. This is critical for supervised learning and evaluation.
2. **Text Data Alone is Sufficient for Classification**
3. The model will rely solely on the textual content (headline and/or article body) to detect fake news. Metadata like user engagement, source credibility, or publication timestamp will not be used unless explicitly included later.
4. **English Language Content**
The initial scope is limited to news articles written in English. Models, preprocessing steps, and embeddings will be optimized for English NLP tasks.
5. **Binary Classification Task**
The fake news detection task is treated as a **binary classification**: news is either *fake* or *real*. Multi-class or nuanced categories (e.g., satire, bias, clickbait) are out of scope for the initial version.
6. **Pre-trained Models Generalize Well to the Task**
It is assumed that publicly available pre-trained transformer models (like BERT or RoBERTa) can be fine-tuned effectively for fake news detection, without requiring massive retraining from scratch.

8. Appendix

- **Abbreviations**

Abbreviation	Full Form	Explanation
Kaggle	-	A platform for data science competitions, datasets, and collaboration.

LIAR	LIAR Dataset	A benchmark dataset for fake news detection (contains labeled short statements).
TF-IDF	Term Frequency–Inverse Document Frequency	A statistical measure to evaluate the importance of a word in a document relative to a corpus.
BoW	Bag of Words	A text representation method where text is converted into word count vectors.
Word2Vec	-	A technique to embed words into vector space using neural networks. Developed by Google.
GloVe	Global Vectors for Word Representation	A word embedding technique developed by Stanford that uses global word co-occurrence statistics.
BERT	Bidirectional Encoder Representations from Transformers	A transformer-based model by Google for NLP tasks, considers context from both directions.
RoBERTa	Robustly Optimized BERT Approach	An optimized and improved version of BERT by Facebook AI.
DistilBERT	Distilled BERT	A smaller, faster, and lighter version of BERT with comparable performance.
ISOT	ISOT Fake News Dataset	A dataset from the University of Victoria (Information Security and Object Technology Lab) for fake news detection.
EDA	Exploratory Data Analysis	The initial investigation of data to discover patterns, spot anomalies, and test assumptions.
LSTM	Long Short-Term Memory	A type of RNN (Recurrent Neural Network) capable of learning long-term dependencies.
BiLSTM	Bidirectional LSTM	An LSTM that processes input in both forward and backward directions.
HuggingFace	-	A company and open-source library known for providing pre-trained NLP models like BERT, GPT, etc.
ROC	Receiver Operating Characteristic	A graphical plot to evaluate the performance of a binary classifier.
API	Application Programming Interface	A set of rules and protocols for building and interacting with software applications.
joblib	-	A Python library used to efficiently serialize and deserialize (save/load) large objects.

pickle	-	A Python module used to serialize and deserialize Python objects.
torch.save	-	A PyTorch function to save tensors or models.
HTML	HyperText Markup Language	Standard language for creating web pages.
CSS	Cascading Style Sheets	Used to style and layout web pages.
UI	User Interface	The point of interaction between the user and a digital product/system.

- **literature reviews**

1. **Fake News Detection Using Deep Learning: A Systematic Literature Review:**

- 1.1. **Researcher Name:**

- Mohammad Alnabhan
- Paula Branco

- 1.2. **Methodology:**

The authors conducted a **systematic literature review** following structured steps:

- They defined inclusion criteria such as:
 - Only **peer-reviewed** journal and conference papers.
 - Focused on **Computer Science** publications.
 - Related to **fake news detection (FND)**.
 - They also applied **backward snowballing** by reviewing references of selected papers to find additional relevant studies.
- For **data extraction**, they used **Covidence**, a web-based tool, and exported the extracted information into a spreadsheet. The extracted features included:
 - Publication date and publication type
 - Classifier/model used
 - Network architecture
 - Dataset used
 - Transfer learning techniques
 - Handling of class imbalance
 - Performance results (accuracy, precision, recall, F-measure, etc.)
- They initially retrieved 1,642 articles, discarded 436 duplicates, screened titles/abstracts and retained 393 papers, and after full-text screening, narrowed it down to 178 research papers for the final analysis.

- 1.3. **Data Set:**

While their study focuses on the reviewed literature (i.e., they did not create a new dataset), they examined **datasets used within the surveyed fake news detection studies**. Key datasets frequently utilized include:

- **LIAR** – ~12.8K short, labeled political statements (multi-class labels).
- **ISOT** – ~45K news articles (binary: fake/real).
- **PHEME** – ~5.8K tweets labeled as false, true, or unverified.
- **FakeNewsNet** – Comprises GossipCop, PolitiFact, BuzzFeedNews (~19,838 items total).

1.4. Result:

From their systematic analysis, the authors identified several key findings:

- **Deep Learning architectures:**
 - **CNN** and **Bi(LSTM)** were among the most popular and effective, with consistently high detection accuracy.
 - **RNN variants** (LSTM, GRU, BiLSTM, etc.) generally showed strong performance (accuracy up to ~99%).
 - **Graph Neural Networks (GNNs)** were less common and generally underperformed compared to CNN and RNN-based models.
 - **Attention-based models** and **BERT-based architectures** showed increasing use and effectiveness, with top accuracy records around 98–99%.
- **Transfer learning** improved detection performance—including pre-trained models such as AlexNet, BERT, etc., with substantial gains in accuracy (e.g., +23.1% improvement using AlexNet).
- **Class imbalance handling** (oversampling, SMOTE, undersampling, cost-sensitive loss) was explored in a subset of studies, with oversampling and SMOTE particularly effective in boosting performance (e.g., ~95–98% accuracy). However, only a small fraction of reviewed papers addressed this challenge.
- The study highlights **key challenges** and gaps: **limited exploration of transfer learning and class imbalance**, dataset diversity issues, and generalization concerns across datasets.

2. A Comprehensive Analysis of Fake News Detection Models: A Systematic Literature Review and Current Challenges:

2.1. Researcher Name:

- Alok Mishra
- Halima Sadia

2.2. Methodology:

Scope and Focus: The authors systematically analyzed how **machine learning (ML)** and **deep learning (DL)** algorithms have been applied for fake news detection (FND), especially focusing on datasets, simulation platforms, and performance indicators. They classified methods and examined their performance in a comprehensive review framework

2.3. Data Set:

Dataset Types Covered: The review highlights a diverse range of datasets used in FND model evaluations, including:

- Political-statement datasets like **PolitiFact**, **LIAR**, and **Weibo**.
- Article-based datasets such as **FNC-1**.
- Social media data from sources like **Twitter** (including BuzzFeed, ISOT), **Weibo**, **FakeNewsNet**, and other **Kaggle** datasets.
- COVID-19 related datasets.
- Customized or self-constructed datasets by researchers.

2.4. Result:

Algorithms & Techniques

- **NLP & Feature Extraction:**
 - Common feature extraction includes n-grams, TF-IDF, stylistic/linguistic features, and word embeddings such as **BERT**, **GloVe**, and Doc2Vec
- **Machine Learning Models:**
 - Supervised models include SVM, hybrid SVMs, Naive Bayes (Bernoulli), decision trees, and ensemble classifiers.
 - Deep learning models include:
 - **RNNs, CNN-LSTM, GRUs, GNNs**, and transformer-based models like **BERT**.
 - Pros and cons of each model type are discussed, such as RNNs capturing sequential data but being resource-intensive, while CNN-LSTM offers strong performance but may lose information depending on feature extraction
- **Architectural Framework:**
 - The paper introduces a generalized FND model architecture handling multimodal input (text, images, audio, video), with steps like data ingestion, feature extraction, representation, fusion, machine learning training, evaluation, and final classification output
- **Performance Metrics:**
 - Commonly used evaluation metrics: **accuracy, precision, recall, F₁-score**.
 - Other metrics include confusion matrix elements (FP, FN, TP, TN), and statistical tests like t-tests, Holm, Nemenyi, Wilcoxon, McNemar's test, and cross-entropy loss
- **Applications:**
 - FND models are commonly applied in domains like **politics, health (e.g., COVID-19), education, e-commerce, tourism**, and **cultural content**

3. Fake News Detection Using (NLP):

3.1. Researcher Name:

- Yaswanthraj S
- Pavithra Kumar Nadar
- Praveshika P
- Preethika KG
- Sanchana NM

3.2. Methodology:

- The system follows a layered architecture:
 - **Data Preprocessing:** text cleaning (normalization, lemmatization, entity recognition), image preprocessing (contrast enhancement, segmentation), and feature engineering (n-grams, POS, rhetorical markers, emotional trajectory).
 - **Analysis Modules:**
 - *Linguistic analysis:* stylometry, credibility markers, deception indicators.
 - *Machine Learning ensemble:* Logistic Regression, Random Forest, Naive Bayes, XGBoost, combined with stacked generalization.
 - *Deep Learning:* BERT-based fine-tuned model, CNN for visual analysis, cross-modal attention.
 - *Fact Verification:* claim extraction, credibility scoring, and source analysis.
 - **Integration Layer:** combines models via dynamic weighting, Bayesian calibration, and uncertainty quantification.
 - **Explainability Module:** feature attribution, counterfactual explanations, confidence visualization.

3.3. Data Set:

- Dataset comprises **verified fake and real news articles**.
- Also mentions the **LIAR dataset** (Wang, 2017) with **12.8K human-labeled short statements** containing metadata, used widely for benchmarking fake news detection models

3.4. Result:

- **Individual Models:**
 - Logistic Regression: Accuracy **77.8%**, stronger on real news (F1-score 0.82) than fake news (F1-score 0.72).
 - Random Forest: Accuracy **79.3%**, more balanced across classes (F1: real 0.82, fake 0.75).
 - Naive Bayes: Accuracy **78.8%**, F1: real 0.82, fake 0.73.

- **Ensemble Model:** Best performance with **81.1% accuracy**, F1-score 0.84 (real news) and 0.76 (fake news).
- **Confusion Matrix:** Out of 212 test samples → 65 true negatives (fake news), 107 true positives (real news), 22 false positives, 18 false negatives.
- **Key Finding:** Ensemble approach outperforms individual models, but class imbalance (more real news than fake) affects detection quality. The model is more effective at identifying real news than fake.

4. Detecting Fake News using Machine Learning: A Systematic Literature

Review:

4.1. Researcher Name:

- Alim Al Ayub
- Ayman Aljarboun
- Praveen Kumar
- Myung Suh Choi

4.2. Methodology:

- The study uses a **Systematic Literature Review (SLR)** approach.
- Papers were collected from databases including **Clarivate Analytics (WoS)**, **ACM Digital Library**, **IEEE Xplore**, and **Elsevier (Scopus)**.
- **Inclusion criteria:** English-language papers, full access available, focus on machine learning and fake/false news detection.
- **Exclusion criteria:** Non-English papers, incomplete access, or unrelated to ML and fake news detection.
- **Selection process:** From 73 papers initially identified, **26 papers** were included after filtering.
- Quality was assessed based on the relevance and contribution of the research regarding ML for fake news detection

4.3. Data Set:

- The review discusses that classifiers are trained using **labeled datasets** (training and test splits).
- Common feature extraction methods mentioned: **TF-IDF**, **N-Gram Model**, **Bag of Words**.
- Highlights that **imbalanced datasets** are a common problem (more real news than fake).
- Specific dataset examples referenced:
 - **LIAR dataset** (Wang, 2017) with human-labeled statements.
 - Social media datasets like **Facebook posts** (Granik & Mesyura, 2017).
 - Other public datasets used in different studies reviewed.
- No single dataset was used directly by the authors since this is a literature review — instead, they compared datasets across the 26 included studies.

4.4. Result:

- **Why ML is required:** Human detection of fake news is difficult; ML allows **automated, scalable, and faster detection**.

- **Effective supervised classifiers:**
 - **Naïve Bayes** – Accuracy up to **96%** in some studies.
 - **Support Vector Machine (SVM)** – Frequently reported as one of the best performers.
 - **Logistic Regression** – Reported with high accuracy in multiple studies.
 - **Decision Tree, Random Forest, KNN, Neural Networks, RNN** – also widely applied.
- **Training process:** Classifiers are trained using feature extraction models (TF-IDF, Bag of Words, N-Grams). Preprocessing steps include stop-word removal and stemming.
- **Findings:**
 - Accuracy varies depending on dataset and algorithm, but **SVM, Naïve Bayes, and Logistic Regression** often outperform others.
 - Training quality and balanced datasets are critical for better results.
 - Some studies showed Naïve Bayes with **74% accuracy** (on Facebook posts), while others reported significantly higher results (90–96%).
- **Key limitation:** Availability of high-quality, balanced **labeled datasets** remains the biggest challenge

9. References

Research Name	References
Fake News Detection Using Deep Learning: A Systematic Literature Review	Reference
A Comprehensive Analysis of Fake News Detection Models: A Systematic Literature Review and Current Challenges	Reference
Fake News Detection Using (NLP)	Reference
Detecting Fake News using Machine Learning: A Systematic Literature Review	Reference