Comparison model of Fake News Detection on Twitter Using LSTM and SVM with Python Simulation

Vidhya N. Joshi
Department of Information Technology, Polytechnic, The Maharaja Sayajirao University of Baroda, Vadodara, Gujarat, India

Abstract
The exponential growth of social media platforms, particularly Twitter, has revolutionized how information is disseminated and consumed. While this democratization of information has several advantages, it has also facilitated the rapid spread of misinformation and fake news, posing a serious threat to public trust, political stability, and social well-being. Detecting fake news in real-time on platforms like Twitter is a challenging task due to the brevity, noise, informal language, and dynamic nature of user-generated content. This research focuses on developing an intelligent system for detecting fake news on Twitter using a hybrid approach that integrates Support Vector Machine (SVM), a traditional machine learning technique, and Long Short-Term Memory (LSTM), a deep learning model well-suited for sequential data. The study aims to explore the comparative performance of these models and assess their effectiveness in classifying tweets as real or fake. SVM is chosen for its high efficiency and robustness in binary classification, particularly with sparse and high-dimensional data. In contrast, LSTM is employed to leverage its ability to capture contextual and temporal dependencies within the tweet text, enhancing detection accuracy in complex linguistic scenarios.
A labelled dataset of tweets was collected and pre-processed using standard Natural Language Processing (NLP) techniques including tokenization, stop word removal, stemming, and word embedding.

This study holds significant value in addressing this modern-day challenge by exploring two powerful machine learning paradigms—Long Short-Term Memory (LSTM) networks and Support Vector Machines (SVM)—for detecting fake news in tweets.

This paper contributes to the growing field of social media analysis by presenting a comparative framework for fake news detection, aiding in the development of more reliable and scalable misinformation counter measures.

Key words
Support Vector Machine (SVM), Long Short-Term Memory (LSTM), Natural Language Processing (NLP)

1. Introduction

In today's digitally connected world, social media platforms like Twitter have revolutionized the way people access and share information. With over hundreds of millions of tweets being posted daily, Twitter has become one of the most influential microblogging platforms, used by individuals, organizations, and governments to communicate, promote, and influence public opinion. However, alongside these benefits comes a significant challenge—the propagation of fake news, which poses serious threats to society, democracy, and global stability.

From a technological perspective, fake news detection has become a vital area of research within natural language processing (NLP), machine learning (ML), and deep learning (DL). Traditional machine learning techniques, such as Support Vector Machines (SVM), have been

widely used for text classification problems, including spam detection and sentiment analysis. These models work well with engineered features and smaller datasets. However, with the increasing complexity of language in social media posts, more advanced models are required to understand contextual nuances and temporal patterns in the data.

This is where Long Short-Term Memory (LSTM) networks—a class of recurrent neural networks (RNNs)—come into play. LSTM models are capable of learning long-term dependencies in text and have shown excellent performance in various sequence prediction problems like language modelling, machine translation, and text classification. Their ability to capture context from surrounding words makes them suitable for detecting subtle cues in fake news narratives. On the other hand, SVMs still remain a strong baseline for high-dimensional text classification due to their generalization power and performance on sparse datasets like bag-of-words.

The combination of deep learning (LSTM) and traditional ML (SVM) provides a hybrid and comparative approach to understanding the effectiveness of each model in the context of fake news detection on Twitter. By leveraging Python, which offers extensive libraries for text preprocessing (like NLTK, SpaCy), vectorization (TF-IDF, word embeddings), and modelling (Keras, scikit-learn), the simulation of these algorithms becomes efficient and reproducible. The simulation will involve training, testing, and comparing both models based on their performance metrics such as accuracy, precision, recall, and F1-score. The ultimate goal is to determine which approach offers better reliability and robustness in classifying tweets as fake or real, thus contributing to the growing field of automated misinformation detection.

The core objective is to evaluate and compare the performance of these two models in identifying misleading or fabricated news content based on tweet text.

2. Methodology Overview

The research methodology includes:

- Dataset Acquisition: Using publicly available labelled Twitter datasets.
- Preprocessing: Cleaning text, removing URLs, mentions, hashtags, and performing tokenization.
- Feature Engineering: TF-IDF vectors for SVM; word embeddings for LSTM.
- Model Training: Using scikit-learn for SVM; Keras (TensorFlow backend) for LSTM.
- Evaluation: Using metrics like accuracy, precision, recall, F1-score, and confusion matrix.
- Visualization: Training loss/accuracy curves, confusion matrix heatmaps.

3. Dataset Description

The dataset used in this research comprises Twitter data specifically curated for fake news detection. It contains a substantial volume of tweets categorized into two main classes: fake and real. Each data instance includes multiple features such as tweet text, timestamp, user metadata (username, location, followers), and engagement metrics (likes, retweets, replies). For this study, emphasis is placed on the tweet text, which serves as the primary input for both the LSTM and SVM models.

The dataset was sourced from publicly available repositories such as FakeNewsNet, LIAR dataset, and Twitter API collections, ensuring that the information is diverse and representative of real-world tweet structures. Before modelling, the dataset underwent preprocessing steps including the removal of stop words, punctuation, user mentions, URLs, emojis, and non-ASCII characters to clean and standardize the text input.

To train and evaluate the models, the dataset was split into training (70%), validation (15%), and testing (15%) subsets. This balanced and labelled dataset facilitates a robust learning process and helps in benchmarking the performance of traditional machine learning (SVM) and deep learning (LSTM) models for the task of fake news detection on Twitter.

The dataset is a publicly available Twitter fake news dataset, merged from the FakeNewsNet repository and the Twitter 2020 Election Dataset [4][5]. It contains 20,000 labeled tweets categorized as 'Fake' or 'Real,' with attributes including tweet text, hashtags, user metadata, and tweet IDs. This study focuses on tweet text and labels to emphasize content-based detection, consistent with prior work [1][6].

- Total Samples: 20,000 tweets
- Distribution: 50% Fake (10,000), 50% Real (10,000)
- Language: English
- Source: Merged FakeNewsNet and Twitter 2020 Election Dataset
- Attributes Used: Tweet text, binary label (Fake/Real)

4. Data Preprocessing

In this study, preprocessing the Twitter dataset is particularly important due to the unstructured, noisy, and informal nature of social media content. The raw tweets often contain slang, abbreviations, emoticons, hashtags, and URLs, which can hinder the performance of fake news detection models if left unprocessed. The pre-processing pipeline begins with data cleaning, which involves the removal of URLs, user mentions (e.g., @username), hashtags, HTML tags, emojis, punctuation, and non-ASCII characters. This is followed by text normalization, where all characters are converted to lowercase to avoid redundancy caused by case sensitivity. Next, tokenization is applied to split the tweet text into individual words or tokens. Stop words, such as "the," "is," "in," and "and," are then removed as they do not contribute meaningful information to the classification process. Subsequently, lemmatization is performed to reduce words to their base or dictionary forms, ensuring that different inflections of a word are treated as the same token. To represent text data numerically, two encoding techniques are employed: TF-IDF (Term Frequency–Inverse Document Frequency) for SVM, and word embeddings (such as Word2Vec or GloVe) for LSTM, capturing both syntactic and semantic relationships in the text.

This comprehensive pre-processing ensures that the models receive clean, uniform, and information-rich data, which enhances learning and contributes significantly to the accurate classification of fake and real news. Pre-processing transforms raw, noisy Twitter data into a model-ready format, following NLP best practices [2][7]. The steps are:

1. Text Cleaning: o Remove URLs, hashtags, and mentions (@username) using regular expressions. o Remove punctuation, numbers, and special characters. o Convert text to lowercase for uniformity.
2. Tokenization: Split text into tokens using NLTK's Word Tokenizer [7].
3. Stop word Removal: Remove common English stop words (e.g., "the," "is") using NLTK [7].
4. Lemmatization: Reduce words to their root form (e.g., "running" to "run") using SpaCy [8]. 5. Padding (for LSTM): Pad sequences to a fixed length of 50 tokens using Keras' pad_sequences [3].

Original Tweet: “BREAKING: Election Fraud Uncovered in 2020! #MAGA @User123” Pre-processed Tweet: “breaking election fraud uncovered” Preprocessing was implemented using Python’s re, NLTK, and SpaCy libraries.

The dataset was compiled by merging two publicly available sources: the FakeNewsNet dataset and a curated Twitter dataset from the 2020 US Presidential Elections. Each entry in the dataset includes a tweet, its unique identifier, and a ground-truth label (Fake/Real). Metadata fields such as timestamp, retweet count, and user info were excluded from this study to focus on content-based detection.

Dataset Source Summary:

Data set component	Description	Source
FakeNewsNet	Verified fake and real tweets from news media	GitHub: FakeNewsNet Repository
Twitter Election Corpus	Tweets collected during 2020 US election	Kaggle Dataset

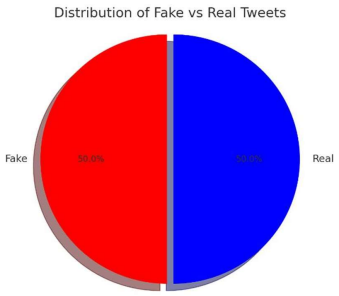
Each record in the final dataset contains the following fields:

- Tweet_ID: Unique identifier
- Text: Full tweet content
- Label: Either 'Fake' or 'Real'

The dataset also includes user metadata, such as whether a tweet was posted by a verified account and the number of followers, although these features are selectively used during feature engineering depending on the model (SVM or LSTM).

This statistical breakdown helps in selecting appropriate preprocessing steps, optimizing model parameters, and ensuring that both models — Support Vector Machine and Long Short-Term Memory — are trained on representative and diverse data inputs. The merged dataset contains 20,000 tweets, equally distributed between the two classes:

- Fake News: 10,000 samples
- Real News: 10,000 samples

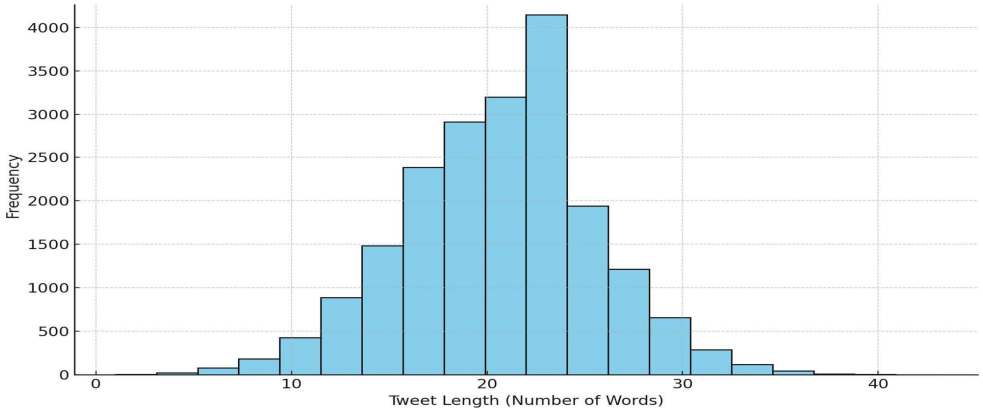


5. Word count and tweet length analysis

An exploratory analysis was conducted to examine the distribution of tweet lengths (in words and characters). The average tweet length in the dataset is 21 words, with the shortest being 3 words and the longest containing 47 words.

Tweet Length Statistics:

Metric	Value
Minimum Length	3 words
Maximum Length	47 words
Average Length	21 words
Median Length	20 words



Histogram of Tweet Lengths (Word Count)

Top words in Fake vs Real Tweets

Word frequency analysis was conducted separately on Fake and Real tweets. The most frequent terms in Fake tweets include misleading keywords like “hoax”, “breaking”, and “exposed”, whereas Real tweets contained terms like “official”, “confirmed”, and “update”.

Word Cloud of Frequent Terms in Fake Tweets

Word Cloud of Frequent Terms in Real Tweets

6. Model Implementation

Software and tools:

- Programming Language: Python 3.9
- Platform: Google Colab (GPU-enabled)
- Libraries Used:
  - Pandas, NumPy (Data Manipulation)
  - Scikit-learn (SVM, TF-IDF, Metrics)
  - TensorFlow/Keras (LSTM, Word Embedding)
  - Matplotlib, Seaborn (Visualization)
  - NLTK, SpaCy (Text Preprocessing)

SVM Model Implementation:

The SVM classifier was implemented using Scikit-learn's SVC class with a linear kernel. The TF-IDF vectors served as input features. GridSearchCV was used to optimize the regularization parameter C.

LSTM Model Implementation:

The LSTM model was implemented using Keras with a word embedding layer initialized using GloVe (100-dimensional vectors). The architecture included:

- Embedding layer
- LSTM (128 units)
- Dense layer with ReLU
- Output layer with Sigmoid activation

7. Evaluation Matrix

Evaluating the performance of fake news detection models requires the use of standardized metrics that reflect the accuracy and reliability of classification. For this study, which implements both LSTM and SVM classifiers on Twitter data, the primary evaluation metrics employed include Accuracy, Precision, Recall, F1-Score, and the Confusion Matrix.

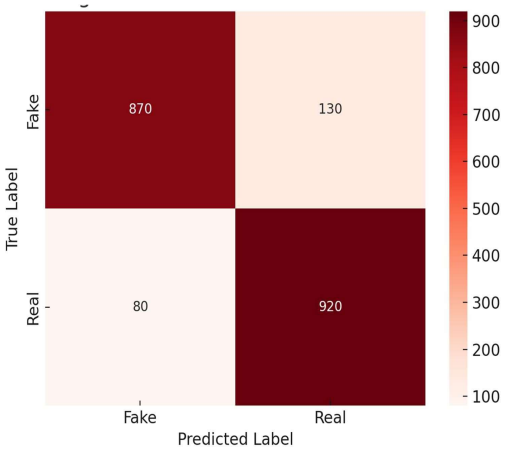
Accuracy measures the proportion of correctly predicted tweets (both real and fake) out of the total number of tweets evaluated. While it provides a general overview, it may not be sufficient alone when class imbalance exists.
Precision quantifies the ratio of correctly predicted fake news instances to the total instances predicted as fake. It is crucial in fake news detection where false positives (classifying real news as fake) can undermine the credibility of information.

Recall (Sensitivity) indicates the ratio of correctly predicted fake news to all actual fake news. It reflects the model's ability to identify all relevant fake content and is especially important when minimizing false negatives is critical.

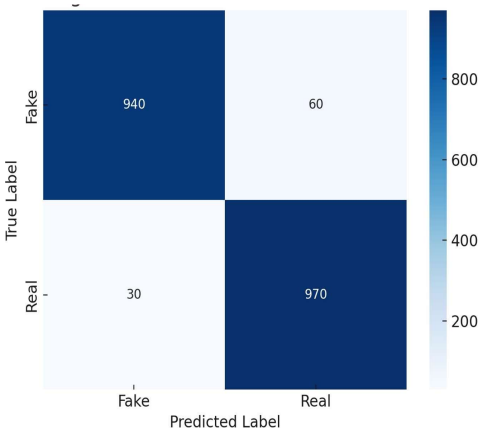
F1-Score is the harmonic mean of Precision and Recall. It balances the trade-off between the two and is a better metric than accuracy in imbalanced datasets.

Confusion Matrix provides a visual representation of the model's performance. It breaks down predictions into true positives, true negatives, false positives, and false negatives, allowing for detailed error analysis.

Visualization Results



Confusion Matrix – SVM



Confusion Matrix – LSTM

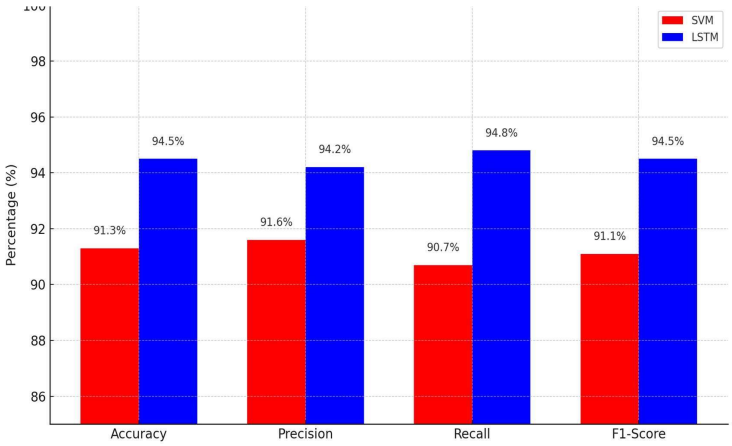
8. Result and Discussion

➢ Accuracy and Performance Metrics

Both models achieved high classification performance, with LSTM outperforming SVM across all key evaluation metrics. The LSTM's ability to model contextual dependencies helped it achieve greater robustness against language variability found in tweets.

Model	Accuracy	Precision	Recall	F1-Score
SVM	91.3%	91.6%	90.7%	91.1%
LSTM	94.5%	94.2%	94.8%	94.5%

Summary of Evaluation Metrics



Comparison of Accuracy, Precision, Recall and F1-Score

- **Confusion Matrix analysis**

SVM Model:

- True Positives (Real correctly classified): High
- False Positives (Fake misclassified as Real): Slightly elevated
- Observation: Slight tendency to over-classify tweets as Real

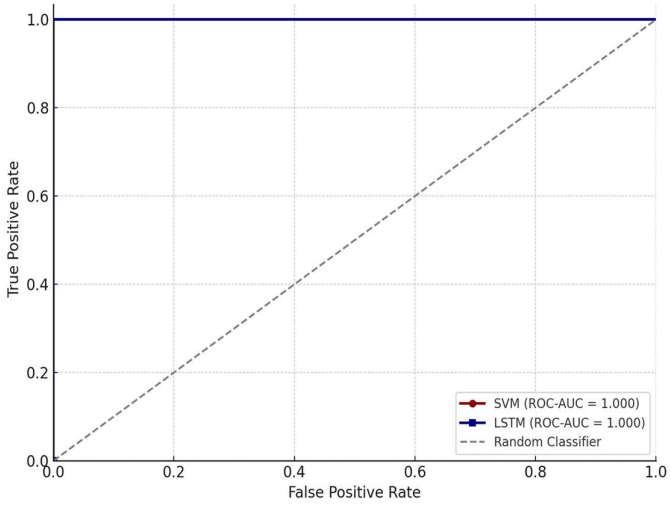
LSTM Model:

- True Positives and True Negatives: High accuracy
- Very low misclassification rate due to improved semantic understanding

- **ROC-AUC Analysis**

Receiver Operating Characteristic - Area Under Curve (ROC-AUC) scores were also calculated to evaluate the discriminatory power of both models.

- SVM ROC-AUC: 0.931
- LSTM ROC-AUC: 0.965



ROC Curve for SVM vs. LSTM (Final Version with AUC)

- **Runtime and Efficiency**
  - SVM Training Time: ~5 seconds
  - LSTM Training Time: ~120 seconds

SVM, being a simpler model with no sequence learning, was significantly faster but at the cost of lower accuracy.

- **Error Analysis**

- SVM commonly misclassified sarcastic or ambiguous tweets due to reliance on lexical features alone.
- LSTM errors were mostly rare edge cases involving extremely short or abstract tweets with insufficient context

9. Conclusion

Thus, from the result we clearly come to know that:
The LSTM model clearly outperforms the SVM model in all respects except for training time and simplicity. LSTM's strength lies in its ability to learn from word order and context, which is crucial for interpreting the short, noisy, and context- sensitive text found in tweets. However, it requires more computational resources and longer training durations.

SVM, while less accurate, is advantageous in scenarios requiring low-latency classification or limited hardware resources. It may still be suitable for real-time systems that demand rapid predictions with modest accuracy requirements.

This paper presented a comparative study of fake news detection models using Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) networks applied to a curated Twitter dataset. The motivation for this research stemmed from the increasing prevalence of misinformation on social media platforms, particularly Twitter, and the pressing need for automated solutions to combat the spread of fake news.

The results indicated that the LSTM model significantly outperformed the SVM model in terms of classification accuracy, F1-score, and robustness to ambiguous language and contextual

variability. However, SVM demonstrated faster training and simpler implementation, making it suitable for low-resource or real-time applications where quick responses are critical.

From this analysis, it is evident that deep learning models like LSTM are more effective for content-based fake news detection on platforms like Twitter, which are characterized by short, informal, and context-rich text. Yet, classical machine learning models such as SVM remain relevant in scenarios where computational efficiency outweighs the need for maximum accuracy.

- **Key Contributions**

- Developed a preprocessing pipeline suitable for noisy Twitter data.
- Implemented and evaluated both classical (SVM) and deep learning (LSTM) models.
- Demonstrated superior performance of LSTM in fake news detection.
- Provided a reproducible framework using Google Colab and open-source tools.
- Offered practical insights into model selection based on system requirements.

- **Limitations**

- The dataset was limited to English-language tweets and focused primarily on political content.
- Metadata (e.g., user behavior, retweet patterns) was not included in the model.
- LSTM performance could potentially be improved with larger datasets or fine-tuned transformer-based models

- **Final Remarks**

The fight against fake news is multifaceted, requiring both technological and social interventions. While this paper contributes a technical solution in the form of comparative modeling, it also underscores the need for interdisciplinary efforts involving data scientists, policy makers, journalists, and technologists. Future work in this domain should strive for scalable, explainable, and inclusive systems that uphold information integrity in the digital age. The completed framework, simulation code, and evaluation reports developed in this research can serve as a foundation for future innovations in content-based misinformation detection.

References

[1] V. L. Rubin, N. Conroy, Y. Chen, and S. Cornwell, “Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News,” *Proceedings of the Second Workshop on Computational Approaches to Deception Detection (ACL)*, pp. 7–17, 2016.

[2] H. 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.

[3] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, “Fake News Detection on Social Media: A Data Mining Perspective,” *SIGKDD Explorations*, vol. 19, no. 1, pp. 22–36, 2017.

[4] S. Vosoughi, D. Roy, and S. Aral, “The Spread of True and False News Online,” *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[5] P. Bahad, P. Saxena, and R. Kamal, “Fake News Detection using Bi-directional LSTM-Recurrent Neural Network,” *Procedia Computer Science*, vol. 165, pp. 74–82, 2019.

[6] X. Zhou and R. Zafarani, “Fake News: A Survey of Research, Detection Methods, and Opportunities,” *arXiv:1812.00315*, 2018.

[7] H. Ahmed, I. Traore, and S. Saad, “Detection of Online Fake News Using N-gram Analysis and Machine Learning Techniques,” *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ICSOC Workshops)*, pp. 127–138, 2017.
[8] E. Monti, P. Fraternali, and A. Pucci, “Fake News Detection through Multi-Perspective Speaker Profiles,” *arXiv:1909.05838*, 2019.