Fake News Detection on Social Networks

Adhithya Sundar[1], Sarang Agarwal[2], Mahesh HB[3]

[1]B. Tech, Computer Science & Engineering, PES University Ring Road Campus Bengaluru, India
Email: *PES1UG19CS026[at]pesuonline.onmicr osoft.com*

[2]B. Tech, Computer Science & Engineering, PES University Ring Road Campus Bengaluru, India
Email: *PES2UG19CS367[at]pesuonline.onmicr osoft.com*

[3]Associate Professor, PES University Ring Road Campus Bengaluru, India
Email: *hbmahesh[at]pes.edu*

Abstract: ***The project titled 'Fake News Detection on Social Networks' aims to combat the rising threat of misinformation in the digital age. With the exponential growth of online content, the spread of false news has become a critical challenge, leading to various social, political, and economic consequences. This project focuses on developing a machine learning-based model that can accurately classify news articles as either true or fake. By employing a combination of Natural Language Processing (NLP) techniques and algorithms such as Recurrent Neural Networks (RNN), K-Nearest Neighbors (KNN), and Long Short- Term Memory (LSTM), the system will analyze the linguistic patterns and contextual information in news content.***

Keywords: Fake News, Machine Learning, KNN, NLP, RNN and LSTM

1. Introduction

Misinformation spreads quickly within the digital landscape, where social media and information circulation has grown faster than in any other time period. Technologies have merged in a way that allows users to consume news instantaneously, but this accessibility comes with the threat of misinformation thanks to fake news, which can completely alter the understanding of the public. It poses a genuine threat, not only to the understanding of the public at large, but also its trust in genuine news. Irrespective of the medium, the impact of fake news can transform regions, from influencing the decisions made by politicians at the top, to altering the general demeanour of the public, as well as reducing faith in all news outlets. In this new digital age, there is a heightened reliance on the internet which has made it difficult for the common user to filter genuineness of news, making it more nuanced than ever. A study from MIT shows that false information now spreads faster than genuine news. Many factors contribute, be it the social media platform or the embedded algorithms. The core of the issue lies in how fake news garners higher reach, with fake news being retweeted 70% more. It reveals how urgent the need to combat the spread of fake news is.

ML has emerged as the perfect tool to address this growing problem of fake news. The algorithms can analyze humongous amounts of data and deduce patterns or linguistic features present within news articles that point out whether they are fake or not. Approaches such as RNN, KNN, and LSTM networks have been adopted to detect fake news, showing promising results. However, these models are not without their limitations, such as challenges with overfitting, generalization, and the inherent biases in training datasets. Moreover, diversity in writing styles, topics, and tones of news articles makes the task of developing a universally applicable detection model complicated.

The importance of an accurate and efficient fake news detection system cannot be overstated, especially in an era where information shapes public opinion and decision-making. For the sake of an effective functioning of democracy and the public at large, sources of news have to be reliable. This paper presents a real-time solution based on machine learning in support of current endeavors to contribute toward fake news detection. By leveraging advanced techniques in natural language processing and machine learning, this approach strives to increase both the accuracy and speed of fake news identification, helping users navigate an increasingly complex media environment with greater confidence.

2. K-Nearest Neighbour Classifier

The K-Nearest Neighbor is probably one of the most widely utilized supervised techniques under machine learning Basically, supervised techniques train models given labeled data points. KNN classification is generally very effective when it comes to anomaly detection - fraud detection applications, for example-and KNN regression has been useful in a great number of applied domains, which include actuarial modeling and environmental forecasting for real estate purposes. Recently, it has also been used for detection of fake news on social media. KNN is a direct approach to creating classification models by allowing the algorithm to assign the class label for an instance as a function of its proximity to other labeled instances. KNN is a family of algorithms rather than a single algorithm, with various algorithms for each task. The algorithm is quite commonly used in recommendation systems and semantic search applications as well as in anomaly detection. It has become popular because of its ease of use, interpretability, and efficiency in noisy data in classification problems in industries.

In machine learning, the first step is to represent the data points as feature vectors, which are mathematical representations of the data. If a dataset has N unique features, the length of the feature vector is N. The objective of the KNN algorithm is to classify a new object based on its attributes and the training data. To classify a new object, KNN follows a series of steps:

1) Compute the distance between the classification object and every point in the training set.
2) Select the K nearest data points based on the smallest distance.
3) Conduct a "majority vote" among the K nearest points to determine the class label of the new object.

Natural Language Processing

Natural language processing (NLP) is an interdisciplinary field that combines computer science, artificial intelligence, and linguistics to explore how computers can understand, interpret, and interact with human (natural) languages.

This method uses the SentiWordNet library, which assigns sentiment polarity values to terms in a document. Each term in WordNet has three scores: obj(t), pos(t), and neg(t), representing objective, positive, and negative polarities, respectively. These scores are calculated using outputs from eight ternary classifiers. WordNet is a comprehensive lexical database of English, where words are grouped into synsets based on their meanings. This approach relies on semantics to analyze language, with key NLP tasks including identifying sentiment-carrying sentence components, understanding sentence structure, and using tools for text processing. The sentiment polarity is determined by comparing the positive and negative scores, with the higher score indicating the sentence's sentiment.

3. Recurrent Neural Networks (RNNs)

A basic approach for text classification tasks is the RNN model designed to handle sequential data. Recurrent Neural Networks (RNNs) is a class of neural networks created for processing sequential data. The RNNs have connections that permit information to be persisted, thereby making them a good fit for tasks such as time series forecasting and natural language processing (NLP). A hidden state that captures information from previous time steps is maintained by RNNs, which facilitates the understanding of sequences. RNNs are particularly appealing to NLP because the presence of a hidden state helps capture information from earlier in the sequence. They suffer from the vanishing gradient problem, meaning they fail to learn from long sequences. Although such a baseline for news article classification is provided, it further attests to the complexities this model faces in handling the intricacies of language and, most predominantly, their failure to digest long-range dependencies as well as contextual details.

Further analysis of the model's performance would show its limitations with regard to precision and recall metrics. The precision was lower than expected, which gave a higher false positive rate; this implies that the RNN may be wrongly classifying some real articles as fake, which is likely to be due to its limited capacity to differentiate subtle differences between classes. Similarly, the recall metric, reflecting the model's effectiveness in detecting actual fake news, was also subpar, meaning that many fake articles went undetected.

4. Long-Short Term Memory (LSTMs)

The Long Short-Term Memory (LSTM) networks are a specific flavor of RNNs designed to handle the vanishing gradient problem. LSTMs are designed using memory cells, which maintain information over long periods. They work well in applications where the model needs to learn long-term dependencies, such as speech recognition and text generation. Using gates controlling the flow of information, LSTMs selectively remember or forget information at each time step. In our implementation, LSTM models were used for fake news detection and showed an improvement over the basic RNN, demonstrating better handling of long-range dependencies and contextual information.

It starts by preparing the text data. This includes tokenizing the text and converting it into sequences of integers. Each word will be assigned an index, and the sequences then padded up to all the same lengths (100 words in this case). This then allows equal input to the model.

The main components forming the model architecture are as follows:

- The Embedding Layer transforms integer sequences into dense word vectors of a fixed size; the dimensions are 128. This aids in maintaining semantic relationships between words.
- The LSTM Layer processes the sequence data, retaining useful information from previous time steps, and this makes it suitable for tasks such as text classification, where the context is important.
- The LSTM layer is set at 100 memory units, and dropout and recurrent dropout help towards further minimizing overfitting.

Lastly, the Dense Layer generates the output, using the sigmoid activation function to classify the input into one of two categories, for instance, "fake" or "real."

This LSTM-based approach has more advantages over other simpler models like traditional RNNs by preventing the vanishing gradient problem and capturing long-range dependencies in data that could be useful to understand the intricate language structures that exist in news articles.

5. Implementation of this Project

1) Long Short-Term Memory (LSTM)

a) Data Collection
- Use datasets like FakeNewsNet, Kaggle Fake News Dataset, or collect data from social media.
- The dataset typically contains text (news articles, headlines, tweets) and labels (real or fake).

b) Data Preprocessing
- Text Cleaning: Remove punctuation, stop words, special characters, and convert text to lowercase.
- Tokenization: Break text into individual words.
- Word Embeddings: Convert words into numerical representations using Word2Vec, GloVe, or TF- IDF.
- Padding Sequences: Ensure all text samples have the same length for processing.

c) LSTM Model Architecture
- Use word embeddings as input.
- Employ a Bidirectional LSTM to capture contextual meaning from both directions.

- Add dropout layers to prevent overfitting.
- Use a sigmoid activation function for binary classification (real or fake).

d) Training and Evaluation
- Train the model using cross-entropy loss and an Adam optimizer.
- Evaluate using accuracy, precision, recall, and F1-score.
- Pros and Cons of LSTM
- Captures long-term dependencies in text
- Effective for complex sentence structures
- Computationally expensive and requires large datasets

2) K-Nearest Neighbors (KNN)

a) Data Preprocessing
- Convert text into numerical representations using TF-IDF Vectorization.
- Normalize the text feature vectors.

b) KNN Classification
- Choose an appropriate value of k (number of nearest neighbors).
- Measure similarity using Euclidean or Cosine distance.
- Assign the class based on the majority label of nearest neighbors.

c) Training and Evaluation
- Train the model on a labeled dataset.
- Evaluate using accuracy, confusion matrix, and F1-score.
- Pros and Cons of KNN
  - Simple and interpretable
  - No training required, just storage of training data
  - Computationally expensive for large datasets
  - Sensitive to noisy data

3) Recurrent Neural Networks (RNN)

a) Data Preprocessing
- Perform text cleaning, tokenization, and embedding (same as LSTM).
- Convert text into word embeddings for numerical processing.

b) RNN Model Architecture
- Use an embedding layer to represent words.
- Pass text through Simple RNN layers to capture sequential dependencies.
- Add dropout layers to reduce overfitting.
- Use a sigmoid activation function for binary classification.

c) Training and Evaluation
- Optimize using cross-entropy loss and Adam optimizer.
- Evaluate using accuracy and F1-score.

d) Pros and Cons of RNN
- Good for processing sequential data
- Captures word order in sentences
- Suffers from vanishing gradient problem
- Less effective than LSTM for long text sequences

6. Graphs of the Implementation of this Project

The following graphs shows that the models had higher true detection rates, with lower accuracy in identifying the true news. After enhancing data cleaning and model tuning, it has come to a point that the models have now higher false detection rates, higher accuracy in identifying the false news. The graphs are shown below:

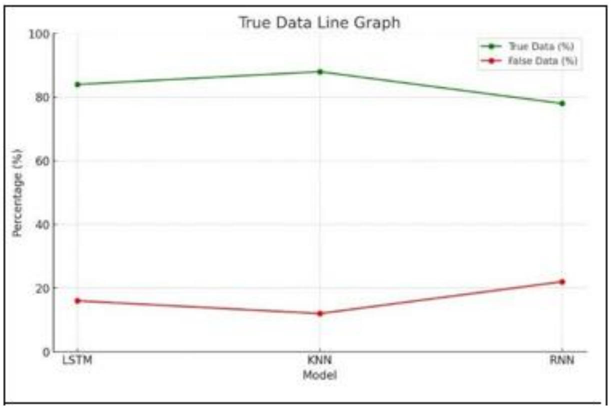

Figure: Incorrect detection of news articles with lower accuracy

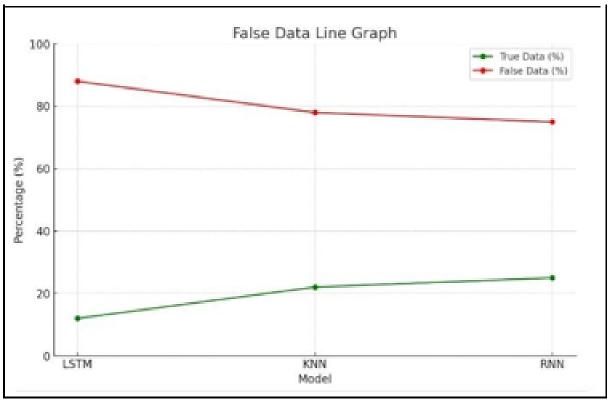

Figure: Correct detection of news articles with higher accuracy

7. Comparisons of each Neural Networks

1) RNNs (Recurrent Neural networks)
- Type: Neural network
- Purpose: RNNs are designed for sequential data. They are used for tasks like time series prediction, speech recognition, and natural language processing.
- How it works: RNNs have loops in their architecture that allow information to persist, meaning they can maintain a “memory” of previous inputs. They take one input at a time, process it, and pass the information along with the next input.
- Limitations: Vanilla RNNs struggle with long-term dependencies because of issues like vanishing gradients, making it hard to retain information over many steps.

2) KNN (K-Nearest Neighbors)
- Type: Instance-based learning algorithm (non-parametric)
- Purpose: KNN is primarily used for classification and regression tasks. It makes predictions based on the

closest training examples in the feature space.

- How it works: Given a new data point, KNN looks at the "K" closest labeled data points and assigns the most common class (for classification) or the average of the values (for regression).
- Key feature: KNN is lazy in that it doesn't learn a model beforehand. Instead, it memorizes the training dataset and performs calculations during prediction time.
- Limitations: KNN can be computationally expensive for large datasets, especially as the dimensionality (number of features) increases.

3) LSTM (Long Short-Term Memory)

- Type: Specialized type of RNN
- Purpose: LSTMs are designed to address the limitations of traditional RNNs, specifically the problem of vanishing gradients. They are used for tasks involving long-term dependencies in sequences, such as language modeling, translation, and speech recognition.
- How it works: LSTMs have a more complex architecture with gates (input, forget, and output gates) that control the flow of information, allowing the network to remember or forget information over long sequences.
- Key feature: LSTMs can retain long-term dependencies, which makes them far superior to vanilla RNNs for tasks that require memory of past events over many time steps.
- Limitations: LSTMs are computationally intensive and require more training data compared to simple RNNs.

4) Key Differences:

- Model Type: RNN and LSTM are neural networks, while KNN is a non-parametric, instance-based learning algorithm.
- Purpose: RNNs and LSTMs are used for sequential or time-dependent data, while KNN is more commonly used for tasks like classification or regression based on feature similarity.
- Architecture: RNNs are simpler, while LSTMs have a more sophisticated architecture with gates that allow for better memory handling. KNN doesn't involve training a model in the traditional sense.

In short, LSTM is usually the better choice over a regular RNN. KNN, on the other hand, is more of a general- purpose classifier or regressor based on proximity, not sequence data.

8. Datasets for this Project

The dataset consists of two types of articles: "fake" and "real" news. It was collected from actual sources from the real world. The true news was gathered from Reuters.com, a widely known news website. On the other hand, the fake news was collected from a number of unreliable websites, found and marked by fact-checking organizations like Politifact (located in the USA) and Wikipedia. The scope is a wide array of topics, but predominantly in politics and world news.

The dataset comprises two CSV files. The first file is titled "True.csv." It comprises more than 12,600 articles from Reuters. The second file is titled "Fake.csv," with more than 12,600 articles from different fake news outlets. In each article, one can find information regarding the title, body text, type (whether it is fake or real), and the date of publication. We mainly focused on articles from 2016 and 2017 in order to align with the fake news data available on Kaggle. Although we cleaned and processed the data, we deliberately left the punctuation and errors in the fake news articles.

9. Results of the Project

In our implementation, the RNN model achieved an accuracy of approximately 75%, the LSTM model achieved an accuracy of around 88% and the KNN model achieved an accuracy of approximately 78%. The comparative analysis of the three models reveals valuable insights into their respective strengths and weaknesses. The LSTM model stands out as the most effective choice for the fake news detection task, with superior accuracy and better performance in both precision and recall metrics. The underlying architecture of the LSTM, which enables it to learn long-term dependencies and contextual relationships within the text, plays a crucial role in its success.

Outcomes of this project have important implications for the development of fake news detection systems. The success of the LSTM model calls for utilizing advanced deep learning architectures that can comprehend the complexity of human language. This finding aligns with recent trends in NLP, where models such as LSTM and other transformer- based architectures have gained wide usage due to their superior performance in diverse tasks of language.

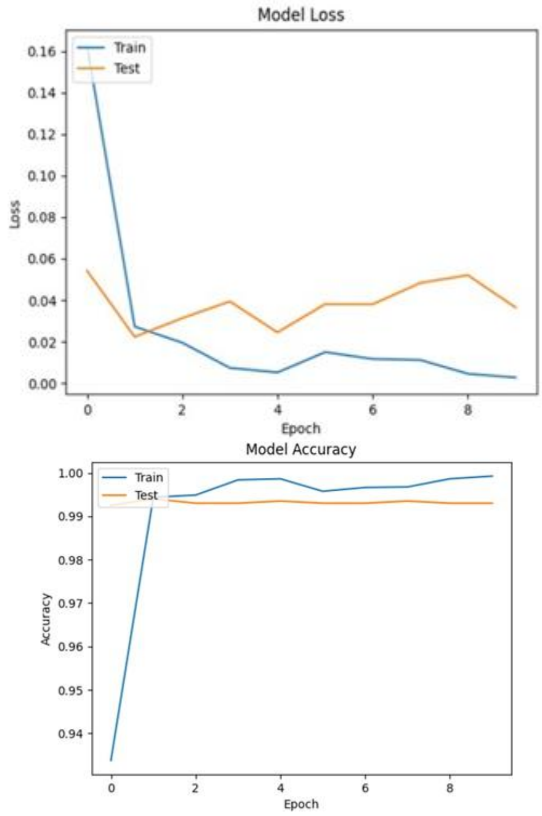

Figure: Training and validation accuracy for the RNN Model

10. Conclusion and Future Work

The conclusion of the project on fake news detection using

Paper ID: MR25514154043 DOI: https://dx.doi.org/10.21275/MR25514154043

machine learning techniques encapsulates the significant findings, contributions, and implications of the research conducted. This project aimed to address the growing concern surrounding the dissemination of fake news and its potential detrimental effects on society. By leveraging advanced machine learning algorithms, particularly Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), and K-Nearest Neighbors (KNN), the project developed a reliable system capable of classifying news articles as either real or fake. The results obtained indicate the effectiveness of these models in tackling the challenges posed by misinformation, with the LSTM model emerging as the most proficient among the algorithms tested.

The comparative analysis of the three models revealed notable differences in their performance metrics. The LSTM model significantly outperformed both the RNN and KNN models in terms of accuracy, precision, and recall. This success can be attributed to the LSTM's ability to capture long-term dependencies and contextual relationships within the text, which are essential for accurately interpreting the meaning of news articles. The results of the project demonstrate that employing sophisticated deep learning architectures is crucial for addressing complex tasks in natural language processing, such as fake news detection.

The project on fake news detection using machine learning has laid a strong foundation by implementing algorithms like RNN, LSTM, and KNN. However, several opportunities for future enhancements can be explored to further improve the accuracy, efficiency, and robustness of the system.

References

[1] Twitter Spam Detection: A Systematic Review: Sepideh Bazzaz Abkenar, Mostafa Haghi Kashani*, Mohammad Akbari, and Ebrahim Mahdipour.

[2] A Survey of Spam Detection Methods on Twitter: Abdullah Talha Kabakus, Resul Kara, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 3, 2017

[3] Spam Detection and Spammer Behaviour Analysis in Twitter Using Content Based Filtering Approach: B. Mukunthan and M. Arunkrishna 2021 J. Phys.: Conf. Ser. 1817 012014

[4] Twitter Spam Detection Using Machine Learning: Asfa Falak, Dr. Hamid Ghous, Dr. Mubashir Malik, International Journal of Scientific & Engineering Research, Volume 12, Issue 2, February-2021, ISSN 2229-5518

[5] Twitter Spammer Detection: Ashwini Bhangare, Smita Ghodke, Kamini Walunj, Utkarsha Yewale, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03 | Mar- 2018

[6] Spammer Detection and Fake User Identification on Social Networks: FAIZA MASOOD, GHANA AMMAD, AHMAD ALMOGREN ASSAD ABBAS ,(Senior Member, IEEE), HASANALI KHATTAK ,

[7] (Senior Member, IEEE), IKRAM UD DIN, (Senior Member, IEEE), MOHSEN GUIZANI, (Fellow, IEEE), AND MANSOURZUAIR,

[8] IEEE Access, Digital Object Identifier 10.1109/ACCESS.2019.2918196

[9] Spam Detection and Fake user Identification Methodologies in Social Networks using Extreme Machine learning: Dr. P. Muthi Reddy,

[10] Kolluru Venkatesh, Deekonda Bhargav, Mallela Sandhya

[11] Survey of review spam detection using machine learning techniques: Michael Crawford*, Taghi M. Khoshgoftaar, Joseph D. Prusa, Aaron

[12] N. Richter and Hamzah Al Najada, Journal of Big Data, Crawford et al. Journal of Big Data (2015) 2:23 DOI 10.1186/s40537-015-0029-9

[13] Fake News Detection Using Machine Learning Approaches: IOP Conf. Series: Z Khanam, B N Alwasel, H Sirafi and M Rashid, Materials Science and Engineering 1099 (2021) 012040 IOP Publishing doi:10.1088/1757-899X/1099/1/012040

[14] Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges: NaeemAhmed, Tariq Shah, RashidAmin, Hamza Aldabbas, Deepika Koundal, BaderAlouf, Wiley Hindawi, Hindawi Security and Communication Networks Volume 2022, Article ID 1862888, 19 pages https://doi.org/10.1155/2022/1862888

[15] Detection and visualization of misleading content on Twitter: Christina Boididou · Symeon Papadopoulos Olga Papadopoulou ·Yiannis Kompatsiaris · Markos Zampoglou ·Lazaros Apostolidis, International Journal of Multimedia Information Retrieval

[16] Automatically Identifying Fake News in Popular Twitter Threads: Cody Buntain, Jennifer Golbeck, arXiv:1705.01613v2 [cs.SI] 30 May 2018

[17] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang. Call attention to rumours: Deep attention-based recurrent neural networks for early rumour detection. arXiv preprint arXiv:1704.05973, 2017.

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

[19] EANN: Event Adversarial Neural Networks for Multi-Modal.

[20] Fake News Detection on Social Media: A Data Mining Perspective Kai Shuy, Amy Slivaz, Suhang Wangy, Jiliang Tang \, and Huan Liuy Aug- Dec 2022 Dept. of CSE Page 63 Fake News Detection on Social Networks.

[21] J. D'Souza, "An Introduction to Bag-of-Words in NLP," 03 04 2018. [Online]. Available:https://medium.com/greyatom/an-introduction- tobag-of-words-in nlp-ac 967d43b428.

[22] G. Bonaccorso, "Artificial Intelligence – Machine Learning DataScience,"10062017. [Online]. Available:https://www.bonaccorso.e u/2017/10/06/ mlalgorithms-addendu m-passive-aggressivealgorithms