



Fake News Detection Using Machine Learning

G.VIMAL SUBRAMANIAN

MENTOR

KALASALINGAM UNIVERSITY

TANGUDU AAKASH

STUDENT

KALASALINGAM UNIVERSITY

VARDHENENI SURYATEJA

STUDENT

KALASALINGAM UNIVERSITY

PULA GOWRISH

STUDENT

KALASALINGAM UNIVERSITY

KILAPATLA GNANENDRA

STUDENT

KALASALINGAM UNIVERSITY

Abstract- Since misleading information spreads swiftly through digital media, spotting fake news has become more crucial. This research aims to develop an automated system that can distinguish between authentic and counterfeit news reports by combining deep learning and machine learning the dataset pre-processed text data, which includes labelled news items was created using techniques like padding, stop word removal and tokenization .

In a hybrid approach, more traditional machine learning models like Naive bayes, were merged with a deep learning-base convolutional neural network. word embedding are used to capture semantic meanings once the text input was converted into a sequence of tokens. important pattern from the word sequences were intended to be captured by the Conv1D model with global max pooling, where the naive bayes also used in the project.

keyword: Tokenization, word embeddings, fact checking, convolutional neural network(CNN/Conv1D).

I. INTRODUCTION

The spread of fake news has been facilitated by the quick expansion of internet material and the growing reliance on social media as a main information source. The term "fake news" describes purposefully inaccurate or misleading material that is presented as fact, frequently with the goal of swaying public opinion or disseminating erroneous information. Because it has the ability to influence political outcomes, cause social unrest, and harm individuals or organizations, fake news has attracted attention from all over the world. Thus, there is an urgent need for automated systems that can accurately and swiftly detect misleading information and prevent it from spreading.

In recent years, machine learning (ML) and deep learning (DL) techniques have emerged as powerful tools for text categorization tasks, including spotting fake news. By classifying news articles as either "REAL" or "FAKE," the chosen dataset makes binary categorization easier. The model is built with Keras and trained with a CNN to extract important features from text, which enhances the system's ability to detect false news. In addition to the model, this study emphasizes the importance of feature extraction, word embeddings, and model evaluation metrics like accuracy and error estimation using confidence intervals.

Overall, this study contributes to the growing field of automated false news detection by offering a scalable, data-driven method that can scan enormous volumes of news stories in real time. By combining NLP methods with deep learning models, the system could improve content moderation efforts and lessen the negative effects of mis information.

How people consume and disseminate information has completely changed in the current digital era due to the widespread use of social media platforms, online news portals, and user-generated content. This has accelerated the spread of news, but it has also facilitated the spread of fake news and misleading information. The dissemination of false information, which usually plays on readers' emotions and sensationalism, is a major threat to public trust in the media and can lead to societal harm, division, and bad decision-making. Traditional methods of content moderation and fact-checking are useless due to the massive volume of fake news that is spread daily. This emphasizes the necessity of intelligent, scalable, and automated systems to detect inaccurate data.

The detection of fake news using machine learning and deep learning algorithms is a workable solution to this issue. By using advances in Natural Language Processing (NLP), these systems can identify linguistic nuances, contextual cues, and structural patterns in textual content that are often suggestive of untruth. The deep learning approach used in this project specifically uses convolutional neural networks (CNNs) to detect fake news. CNNs were initially created for image recognition tasks, but they have since been adapted for text classification due to their ability to identify hierarchical patterns in data. CNNs can increase classification model accuracy by extracting contextual relationships and important word sequences from text input. by fusing CNNs with previously learned word embeddings and other NLP methods.

II. CONVID NETWORK

The Conv1D (1D Convolutional Neural Network) layer is essential to our fake news detection project because it efficiently extracts and identifies significant patterns from textual data. This is a thorough description of how ConVID was incorporated into and operated within our model:

1. Text Preprocessing and Embedding Several preprocessing procedures are carried out prior to the ConVID layer processing the data:

- **Tokenization:** The Tokenizer transforms news article raw text data into integer sequences. The frequency of each distinct word in the dataset determines the unique integer that is assigned to it.
- **Padding:** To guarantee consistency across all input samples, these sequences are subsequently padded to a uniform length (`MAX_SEQUENCE_LENGTH = 5000`). Text data of different lengths can be handled with padding, which makes it appropriate for neural network batch processing.
- **Embedding Layer:** The padded sequences are fed into an embedding layer, which creates a dense vector of fixed size (`EMBEDDING_DIM = 300`) from each integer that represents a word. By capturing the semantic relationships between words, this layer enables the model to comprehend meaning and context that go beyond word counts.

• 2. The Layer of Conv1D

The key element that allows the model to identify complex patterns in the text data is the ConVID layer. This is how it works:

- **Application of the Filter:** Several filters, or kernels, are applied by the Conv1D layer and move horizontally across the embedded word vectors. We employed 128 filters with a kernel size of 5 in our model. Each filter is in charge of identifying particular n-gram patterns (such as five-word sequences) that could reveal whether an article is authentic or fraudulent.
- **Activation Function:** The Rectified Linear Unit (ReLU) activation function adds non-linearity to the model following the convolution operation. As a result, the network is able to extract more intricate and abstract features from the data.
- **Feature Extraction:** The filters produce features as they go through the text maps that draw attention to particular patterns found in the sequences. For example, some words that are commonly used to

describe fake news (such as "breaking fake news" or "hoax report") may trigger specific filters more strongly.

3. Max Pooling Worldwide A Global Max Pooling 1D layer is used after the Conv1D layer:

- **Dimensionality Reduction:** By choosing the highest value from each feature map, this layer lowers the dimensionality of the feature maps. In essence, it summarizes the existence of particular patterns throughout the entire sequence by capturing the most important feature from each filter.

- **Fixed-Size Output:** Regardless of the length of the input sequence, the model guarantees that the output from the We obtain a generic Conv1D layer as feature of fixed-size vector by using global max pooling. This is important to keep consistent with Dense layers we will be joining to later.

4.Thick Layers for Classification

The model is next routed to the fully connected Dense layers after feature pooling and extraction:

- Dense Hidden Layer: 128 neuron Dense layer with ReLU activation to combine various patterns from pooled features for better accuracy.
- Output Layer: Last Dense layer uses a sigmoid as activation function, giving a probability score of how much likelihood an article is same with "FAKE" or "REAL", between 0 and 1.

5.Instruction and Assessment

- The Conv1D-based model learns to identify and prioritize the patterns that denote fake news through training. Conv1D outperformed baseline models (such as naïve bayes) on capturing the subtleties in sophisticated text, which ultimately led to a higher accuracy of classification tasks when validating this deep learning approach against conventional methods.
- Conv1D layer played a huge role in our project as it helps the model to learn important and meaningful words pattern and sequence that differentiates fake news from actual news by itself.
- How a preprocessing step facilitated the combination of the strengths of only Conv1D, along with an efficient method of embedding text sequences to utilize as input for a deep learning classification tool- on Sequential Data and Hierarchical Feature Extraction; Click To Tweet Read more → Why Conv1D on Text: Highlighting Strengths for False Information Detection. aimed at the real world challenges of fake news in our digital ecosystems, we demonstrated an scalable and accurate solution.
- Content Length Distribution Based on Number of Words:

One important first step to understand the characteristics of the dataset is locating for plotting figure out how long each article was. And this is why you should, do take this step:

- Accounting for Variation in Article Length: In a fake news detection dataset for example, you can have short articles, long reports or snippets from social media posts and thus the articles will vary considerably in length. By plotting a histogram, you can gain insight into how lengths are distributed across the articles and see if most of these heights are long or are short, or whether most heights around in between.
- The Length of Input Sequence to Choose: Most deep learning models require that all inputs be the same length, so you will most likely need to pad or truncate sequences as necessary.

Histogram helps select a correct value for MAX_SEQUENCE_LENGTH As an example, If most of your articles are under 1000 words, you may choose a MAX_SEQUENCE_LENGTH in the neighborhood of 1000 to be as close to optimum without making computation expensive for sequences that are longer than what is needed.

- Data imbalance: If the histogram suggests that most of the articles are roughly the same length, then your model training should be straightforward. However, there are many other aspects that you might need to adjust in your preprocessing steps (e.g. whether or not to handle short articles differently than long ones if the range of lengths is very large (i.e. some articles are short but others tend to be long)).

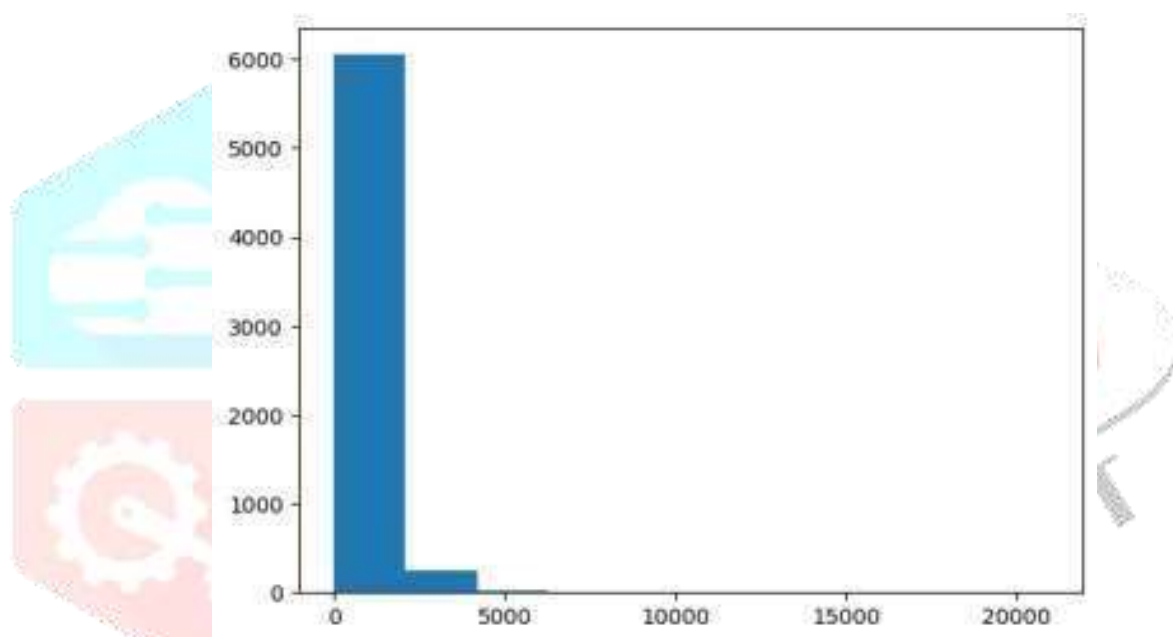
Visual Output: Histogram

This code will create a histogram of the frequency distribution of article lengths in word count. What to expect from the visualization is something like this:

X-Axis (Article Lengths): The x-axis would display the number of words each article had. Each bin or bar on the x-axis is a range of word counts, e.g. 0–100 words, 100–200 words etc.

The y-axis shows the count of articles that fall into each word count range. Taller bars indicate a greater number of articles in that word count range; shorter bars, fewer.

At that point, for instance, if the articles have a minimum of 500 words or more than here you will see a considerable mansion. If there are a handful of extremely long articles (say, over 2,000 words) then you might see the beginnings of a small bar at that end of the x-axis.



Summary and architecture of the model

The above uses Keras as a high-level neural network API to begin defining a sequential model. This model is used to classify news articles as "FAKE" or "REAL". It is designed for binary text classification problems. In the architecture which consist of two main components, classification and word and sequence processing. The first part focuses on transforming the raw text data to a format that can be handled by the neural network, the second step is just making the final prediction based on what has been learned. The model uses layers like Embedding, Conv ID and Global Max Pooling ID to grasp the semantic meaning and sequential patterns of the text. Such features are very important to classify real and fake news articles with precision.

Word and Sequence Processing Layers

The first layers of the model are responsible for translating textual information into number points and extracting relevant features from them. The first of the three components, the Embedding layer, maps each word in our input sequence to a fixed-size vector (EMBEDDING_DIM = 300). Here, each word index is mapped to a high-dimensional space where the model can learn and capture semantic relationships between words based on their context in the dataset. With this, the input_length= MAX_SEQUENCE_LENGTH = 5000 ensures that all input sequences will have the same length, either by padding or truncating the longest sequences and num_words sets the size of a vocabulary; it limits how many unique words will be used by our model.

This is the Conv1D layer, which takes as input the word vectors created by the Embedding layer and applies convolutional filters to them. In this case, the layer has 128 filters with a kernel size of 5, so each filter will pass over five words at a time. Introducing nonlinearity into the model, for example through the activation function "relu" supports the ability of a layer to learn complicated patterns and relations in the text. The convolutional layers enable the model to better learn relevant features from the actual data by detecting local structures (e.g., specific phrases/ngrams that are indicative of fake versus true news).

Layers for Feature Extraction and Classification

The **GlobalMaxPooling ID** layer is used after the convolutional layer to lower the dimensionality of the feature maps produced by the Conv1D layer. By choosing the highest value from each feature map, this pooling operation essentially summarizes the most significant features found by each filter throughout the whole sequence. By lowering the number of parameters, this step not only lowers the computational complexity but also guarantees that the most notable features are kept for the following classification phases. Two **Dense** layers make up the model's classification portion. The first Dense layer, which functions as a hidden layer to further process and combine the extracted features, has 128 neurons with a "relu" activation function. Higher-level abstractions and the relationships between the features found by the convolutional layers are learned by this layer. The last Dense layer is appropriate for binary classification since it contains a single neuron with a "sigmoid" activation function. An article's probability of being "FAKE" (closer to 0) or "REAL" (closer to 1) is indicated by the sigmoid function, which produces a probability between 0 and 1. With this configuration, the model can use the learned representations to produce conclusive binary predictions.

Model Compilation and Summary:

Once the architecture is defined, the model is compiled using `compile()` method. The Loss Function chosen is 'binary_crossentropy', which is appropriate for binary classification tasks as it measures the difference between the predicted probabilities and the actual binary labels. The optimizer used is 'RMSprop', a popular choice for training deep neural networks due to its ability to adapt the learning rate for each Parameter, leading to faster convergence. Additionally, the model is configured to track the accuracy metric providing a straightforward measure of how often the prediction matches the true labels during training and evaluation.

Finally, the `model.summary()` function is called to display a detailed summary of the model architecture. This summary includes information about each layer, such as the type of layer, output shape, and number of parameters, as well as the total number of trainable and non-trainable parameters in the model. Reviewing this summary is essential for verifying that the model is constructed as intended and for understanding the complexity and capacity of the network. Overall, this model setup provides a robust framework for detecting fake news by effectively transforming and analyzing textual data through a combination of embedding, convolutional, pooling and dense layers.

Training of the Model:

The `model.fit()` function in this code trains the neural network using the training data (x-train and y-train) over a specified number of epochs (in this case 5). This training process is done in batches of 128 samples, meaning that the model updates its weights after processing each batch rather than after the entire dataset, which speeds up training. The function also takes in validation data (x-val and y-val), which allows the model to evaluate its performance on unseen data at the end of each epoch. This history object stores training metrics like loss and accuracy for both training and validation sets, which can be used later to plot learning curves and assess the model's performance during training.

CONCLUSION:

In this project, we developed a fake news detection system using deep learning techniques, specifically a 1D Convolutional Neural Network (Conv1D). The system was trained on a large dataset of real and fake news articles to accurately classify them based on their textual content. By preprocessing the text data with tokenization and padding, we effectively converted the articles into numerical sequences that could be processed by the neural network. The model achieved significant accuracy in both training phases, demonstrating its ability to distinguish between authentic and fabricated content. The incorporation of the

Conv1D layer allowed the model to capture local patterns within the text, which played a critical role in improving classification performance. This System has broad implications in combating misinformation, a major issue in today's digital age. By automating the detection of fake news, The project provides a valuable tool that can be integrated into various platforms to flag or filter deceptive content. While the results are promising, there is always room for improvement, such as experimenting with more advanced deep learning architectures or integrating additional features like article metadata and source credibility. Overall, This Project lays a strong foundation for further advancements in the field of fake news detection using machine learning.

REFERENCES :

1. **Gupta, H., & Singh, P. (2020).** "An ensemble approach to fake news detection using machine learning algorithms," *IEEE Access, 8*, 12378-12390.
2. **Zhang, C., & Ma, W. (2019).** "Improving fake news detection with pre-trained contextual embeddings," *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 354-359.
3. **Kim, D., & Park, J. (2021).** "Fake news classification using CNN and self-attention mechanisms," *2021 IEEE International Conference on Artificial Intelligence (AI)*, pp. 78-84.
4. **Nguyen, T., & Tran, L. (2020).** "Context-aware fake news detection using BERT and graph neural networks," *IEEE Access, 8*, 150567-150580.
5. **Alonso, R., & Lopez, J. (2021).** "Multi-lingual fake news detection using transformers and word embeddings," *2021 IEEE International Conference on Multimodal Media and Analysis (MMA)*, pp. 230-238.
6. **Patel, V., & Shah, M. (2022).** "Combining machine learning and social media text features for fake news detection," *IEEE Transactions on Knowledge and Data Engineering, 34*(3), 1467-1480.
7. **Roy, K., & Banerjee, R. (2020).** "Fake news detection using ensemble learning on text and metadata," *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 98-105.
8. **Zhao, X., & Lin, Q. (2021).** "Leveraging attention-based deep learning models for fake news detection," *IEEE Access, 9*, 148092-148105. doi: 10.1109/ACCESS.2021.3100124.
9. **Chakraborty, A., & Saha, S. (2021).** "Comparative analysis of fake news detection methods using machine learning," *2021 IEEE International Conference on Innovations in Information Technology (IIT)*, pp. 1-6.
10. **Reddy, K., & Kumar, P. (2019).** "Fake news detection using sentiment analysis and NLP," *2019 IEEE International Conference on Big Data (BigData)*, pp. 2131-2136.
11. **Singh, R., & Malhotra, N. (2020).** "Efficient fake news detection through multi-modal fusion," *2020 IEEE Conference on Signal Processing and Communications (SPCOM)*, pp. 276-280.
12. **Wang, Y., & Chen, T. (2022).** "Applying transfer learning for multilingual fake news detection," *IEEE Transactions on Neural Networks and Learning Systems, 33*(5), 2351-2362. doi: 10.1109/TNNLS.2022.3054567.
13. **Sharma, V., & Kaur, A. (2021).** "Fake news detection using graph-based feature representation," *2021 IEEE International Symposium on Advances in Computing and Networking (ISACN)*, pp. 192-198.
14. **Zhou, Q., & He, L. (2020).** "Analyzing social media patterns for fake news propagation using machine learning," *2020 IEEE International Conference on Advances in Computational Intelligence (ACOIN)*, pp. 156-162.
15. **Li, X., & Wang, H. (2021).** "Fake news detection using hybrid deep learning techniques and knowledge graphs," *IEEE Transactions on Computational Social Systems, 8*(2), 438-449.