Social Media Fake Information Identification Method Based on LSTM

Shuhan Liu *

School of Mathematics, University of Leeds, Leeds, United Kingdom

* Corresponding author: mm22sl2@leeds.ac.uk

Abstract. False information is spreading like wildfire on social media platforms, causing unnecessary confusion and mistrust. Most people do not have the ability to judge for themselves whether information is true or false that is why it is important to use machine learning models to help people identify false information on the web. LSTM model is a good choice for dealing with identifying social media disinformation. The key steps in identifying social media disinformation using LSTM include data preparation and preprocessing, building a model structure adapted to LSTM, model training and evaluation, and subsequent hyper-parameter tuning and model deployment. These steps constitute a complete process that enables LSTM to effectively identify false information and provide credible prediction results through the processing of text data and model training. After model training and validation, LSTM has an accuracy of 99%, AUC of 100% and successfully identifies false information on social media, providing reliable classification results for both true and false information.

Keywords: Fake Information Identification, LSTM, Natural Language Process.

1. Introduction

In today's digital age, social media has become a major platform for people to obtain information, communicate and express their views. However, it has also given rise to a serious problem: the widespread dissemination of false information. False information, including fake news, rumours and misleading content, not only disturbs the public's perception of the truth, but also has a serious impact on society and individuals. The dissemination of disinformation is often rapid and widespread, leading to dire consequences such as public panic, social instability and crises of confidence. The generation of false information in social media stems mainly from the rapid development of digital technology, which makes it easy for anyone to post information, while the mechanisms for regulating and verifying information are relatively weak. False information spreads quickly and generates widespread attention through means such as evocative headlines, compelling images and the use of trending topics, leading to a blurring of the line between what is true and what is false. This not only affects public trust in news and information, but can also lead to social instability and public panic. In this context, the development of efficient and accurate social media disinformation recognition techniques becomes crucial. By combining natural language processing and machine learning techniques, especially deep learning models such as LSTM, it is possible to distinguish false information from real content from massive social media data. Identifying false information not only helps to protect the public from misinformation, but also maintains the credibility of social media platforms, promotes normal information dissemination, and maintains social harmony. Therefore, solving the problem of social media disinformation is crucial to maintaining a healthy information environment and social stability.

Similar studies have been done before for the identification of fake news on social media. For example, using consumer feedback, Tacchini and colleagues.[1] used a binary network to identify fake news based on consumer feedback; And by using deep neural network models, Zhou[2] and colleagues had analyzed the dissemination patterns of fake news across social networks, by testing it on a real dataset, they obtained an accuracy of 83.5%; Also using a deep neural network model, Shu[3] and colleagues proposed the TriFN model for fake news detection and tested it on two fake news datasets, showing accuracy rates of 86.40% and 87.80%; Wang and his colleagues[4] worked on an

end-to-end framework named Event Adversarial Neural Network (EANN). The experiments were conducted using fake news from Weibo and Twitter as datasets, and the final accuracy rates were 71.50% and 82.70%, respectively; Feng and colleagues [5] explored the application of syntactic stylometry to detect deception. They further employed Context-Free Grammar (CFG) rules to identify deceptive reviews. Their investigations were conducted using a dataset of hotel reviews, resulting in a commendable accuracy of 91.20%; Pérez-Rosas and his colleagues[6] discussed linguistic differences in the content of fake and legitimate news and experimented with language-based aspects, showing that 74% accuracy was achieved using a celebrity news dataset; Yang and his colleagues[7] have developed a new type of convolutional neural network, the TI-CNN, which incorporates basic text and image features for more efficient classification; Gupta and his colleagues[8] proposed two methods, CITDetect and CIMTDetect, and tried SVM, and the accuracy of the final study was at 81.30% and 81.80%; Irawan and his colleagues[10] used traditional methods such as Random Forest, SVM, KNN, decision tree, Naive Bayes and XGboos for social media disinformation detection and discuss their accuracy rates

Finally, this paper used LSTM advanced neural network model to detect disinformation on social media in recent years. Tokenizer is a key tool in LSTM's processing of social media disinformation. It is used to convert textual data into sequences of numbers for input into the LSTM model for processing. The main role of the Tokenizer is to map words in the text to unique integer values, thus creating a vocabulary and generating sequences of numbers to represent each text sample. Through the Tokenizer's transformation, the textual data is represented as a sequence of numbers that can be processed by the LSTM model. This enables the LSTM to effectively learn and understand patterns and relationships in textual data, leading to more accurate identification of disinformation on social media. The application of the Tokenizer helps to transform textual data into an input format suitable for deep learning models, enhancing the performance of the LSTM in disinformation recognition. As a final part, this paper compares the accuracy of LSTM with Random Forest, Naive Bayes and KNN, which are traditional approaches, and the results show that LSTM is the most accurate in dealing with the recognition of false information.

2. Methodology

2.1. Pipeline

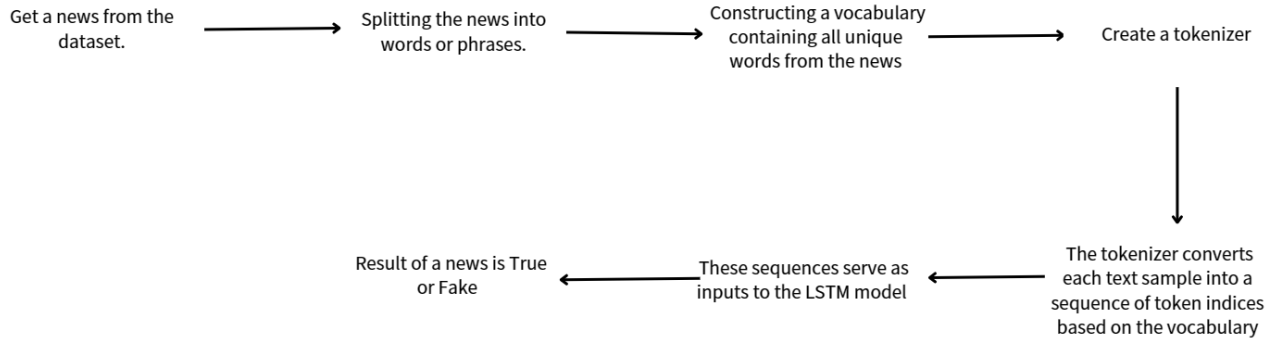

Figure 1. The flow of LSTM to identify the fake news

As shown in the figure 1, the main steps of LSTM model in processing news on social media are firstly labelling fake news as 0 and real news as 1. Then a tokenizer is created to classify the text into words or phrases and then a vocabulary list is created which contains the unique words inside the text. Next, the tokenizer converts each text sample into a sequence of token indices based on the vocabulary. finally these sequences can be imported into the LSTM model and will output the probability of whether the news is true or false.

2.2. Tokenizing Method

Tokenizer is a text processing tool used to convert text data into digital sequences for processing

in deep learning models. Its working process consists of two main phases: fitting and transformation. In the fitting phase, Tokenizer scans the text data and constructs a vocabulary that maps each word to a unique integer identifier. This vocabulary will be used to subsequently transform the text data into a sequence of numbers. In the transformation phase, the Tokenizer converts the words in the text data into corresponding integer identifiers to form a sequence of numbers. This sequence can be used as input to a deep learning model. In addition, in order to maintain a uniform dimension of the input data, the Tokenizer can also pad the end of the sequence so that all sequences have the same length. In summary, the role of the Tokenizer is to transform textual data into a numerical representation for processing in deep learning models, while preserving the semantic information of the textual data. It is an integral part of building a text processing process.

2.3. Long Short-Term Memory

Long Short-Term Memory (LSTM) is a special type of recurrent neural network mainly used to process sequence data. It is able to effectively capture long-term dependencies in sequences for better understanding and prediction of data through mechanisms such as memory units, forgetting gates, input gates, output gates, etc. LSTM can selectively retain or forget information, allowing it to extract important features with excellent performance when processing tasks such as text, time series, and so on.

LSTM highlights its superiority in solving the social media disinformation problem. As a recurrent neural network, it excels in processing sequential data and capturing the temporal dependencies of complex disinformation patterns. It is designed to retain information for multiple time steps, capturing long-term dependencies in disinformation narratives. The LSTM's contextual understanding allows it to discern subtle meanings, automated feature extraction reduces tedious engineering, and adaptability ensures stability in changing data. Its advanced learning capabilities allow it to respond quickly to changes in disinformation dissemination strategies. Taken together, LSTM provides powerful support for social media disinformation recognition with its processing of sequential data, long-term information capture, contextual understanding, automatic feature extraction, adaptability, and learning capabilities.

3. Experiment

3.1. Dataset

There are two datasets available on Kaggle containing social media disinformation and real information from recent years, and the total number of two datasets is 44919. Surprisingly, the data indicates that there are higher numbers of fake news compared to real news, with 21417 real news and 23502 fake news [9]. And word cloud images can visually display the frequently occurring keywords in real and fake news. These keywords may include important information and features that are crucial for the LSTM model, as shown in the figure 2.

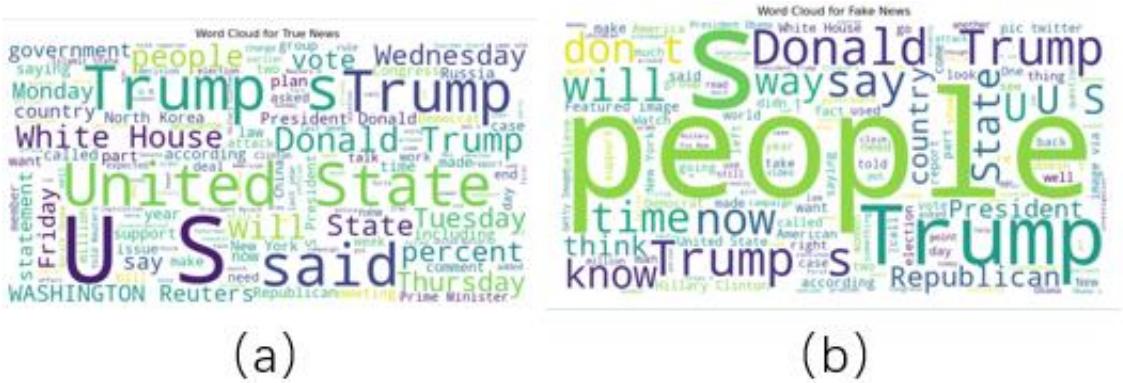

Figure 2. Word Cloud of True and Fake (a)Word Cloud for True News (b)Word Cloud for Fake News

3.2. Data preprocessing

First, saving documents containing true and false news as CSV files, and use the Pandas library to read the CSV files and store the data in the form of Data Frames. Secondly, adding labels to the data. In order to distinguish during the training process, labels of 1 will be assigned to genuine information, while labels of 0 will be attributed to fabricated information. Subsequently, the datasets containing both authentic and fabricated information are merged and their sequence is randomized. This reordering is conducted to streamline ensuing data preprocessing and model training procedures. The data frame as shown in the table 1.

Table 1. Table of True and Fake News

Title	text	subject	date	label
Obama Announces Unfinished Business. For 2...	President Obama began the new year of 2016 wit...	News	1-Jan-16	0
White House says Obama will not discuss FBI pr...	ABOARD AIR FORCE ONE (Reuters) - President Bar...	Politics News	5-Jul-16	1
...	...	...	...	...
44919 rows×5 columns				

3.3. The role of tokenizer

Before initiating training, preprocessing the text data is a crucial step. The process commences with the initialization of a Tokenizer object, which fulfills the role of mapping words within the text data to numerical values, rendering them computationally accessible. The parameter `num_words=5000` is configured to determine the extent of our vocabulary, signifying the inclusion of the 5000 most frequently occurring words from the entirety of the text data into our vocabulary set. Subsequently, the `fit_on_texts` method is employed to provide the Tokenizer object with the text data. This phase is dedicated to constructing the vocabulary, which entails documenting words found in the text alongside their corresponding numerical indices. Following this, the `texts_to_sequences` method comes into play, converting the text data into sequences comprised of numbers. Each word is substituted with its corresponding numerical index from the vocabulary, thereby generating a sequence of numerical sequences. Finally, the `pad_sequences` function ensures uniform sequence length, a prerequisite for machine learning model compatibility. In this context, sequences are either padded or truncated to a length of 200, aligning with the input specifications of machine learning models. The padding procedure employs zero values. Throughout this process, the original text data is transformed into a numerical matrix labeled as X, featuring dimensions of (sample count, sequence length). Each entry constitutes a numerical representation denoting the vocabulary index of a given word from the text. This matrix is eligible for utilization as input to deep learning models, facilitating training and predictive endeavors.

3.4. LSTM model building and random forest, KNN, Naive Bayes models training

Using Keras, construct an LSTM model. First, a sequential neural network model was formulated. In the model, the textual data passes through an embedding layer and is transformed into a vector representation so that it can be understood by the neural network. Next, we added an LSTM layer, which captures sequential patterns in the text and helps to identify patterns of false information. Subsequently, a Fully Connected layer was incorporated to extract more advanced features, followed by the utilization of a Dropout layer to mitigate overfitting concerns. Finally, an Output layer was introduced, employing a Sigmoid activation function to generate output values ranging between 0 and 1, representing the likelihood of false information. In the compilation phase of the model, the chosen loss function was binary_crossentropy, which suits binary classification tasks. The Adam optimizer was employed to adjust model parameters for improved data fitting. Accuracy was selected as the evaluation metric to gauge model performance. The training process involved utilizing the

training datasets (x_train and y_train), where the model underwent training in batches of 64 samples per epoch over a span of 10 epochs. The validation dataset, constituting 20% of the training data, was allocated to assess the model's performance on previously unseen data. The whole process makes the model gradually learn the patterns and features in the data, and by optimising the loss function and parameters, it is expected to be able to identify false information on social media with high accuracy. The performance and training progress of the model will be saved in the history object for further evaluation and analysis.

Next, Naive Bayes, K Nearest Neighbors (KNN) and Random Forest are compared with LSTM models. Random forest, KNN, Naive Bayes and LSTM are common methods for disinformation recognition. Random forest is suitable for a large number of features, KNN is suitable for similarity determination, and Naive Bayes is suitable for textual features. LSTM is suitable for sequential data, captures contextual relationships, and is robust despite requiring more data and computational resources. The building and training of the above four models as shown in the table 2.

Table 2. Table of Construct of LSTM, Naive Bayes, KNN and Random Forest

LSTM	Naive Bayes	Random Forest	KNN
Model construction section: input_dim=5000 output_dim=100 input_length=200 LSTM(128, dropout=0.2, recurrent_dropout=0.2) Dense(64, activation='relu') Dropout(0.5) Dense(1, activation='sigmoid') Model compilation and training section: loss='binary_crossentropy' optimizer='adam' metrics=['accuracy'] Training process: epochs=10 batch_size=64 verbose=1 Training dataset division: test_size=0.2 random_state=42	Feature extraction section: max_features=5000 Dataset division: test_size=0.2 random_state=42 Constructs the plain Bayesian model section: MultinomialNB()Train the model section: nb_model.fit(x_train, y_train).	Feature extraction section: max_features=5000 Dataset division: test_size=0.2 random_state=42 Construct the random forest model section: n_estimators=5 max_depth=5 random_state=42 Train the model section: rf_model.fit(x_train, y_train)	Feature extraction section: max_features=5000. Dataset division: test_size=0.2 random_state=42 Construct the KNN model section: n_neighbors=5 Train the model section: knn_model.fit(x_train, y_train)

4. Results

4.1. Accuracy

The Long Short-Term Memory (LSTM) model has exhibited remarkable performance in the realm of social media fake news detection. With an accuracy rate reaching approximately 99%, and AUC rate reaching approximately 100%. From the below table the accuracy of Random Forest, Naive Bayes and KNN is in the range of 86%-93% and AUC is in the range of 94%-98%. The advantages of the LSTM model in terms of AUC and accuracy imply that it has higher precision and the ability to discriminate between true and false information more accurately in social media disinformation recognition when compared to Random Forests, KNN, and Naive Bayes. This suggests that LSTM provides a more reliable means of information screening and identification by exhibiting better performance in distinguishing between genuine and deceptive content. The accuracy as shown in the table 3.

Table 3. Table of accuracy and AUC

	Random Forest	Naive Bayes	KNN	LSTM
Accuracy	86%	93%	91%	99%
AUC	94%	98%	97%	100%

4.2. Confusion matrix

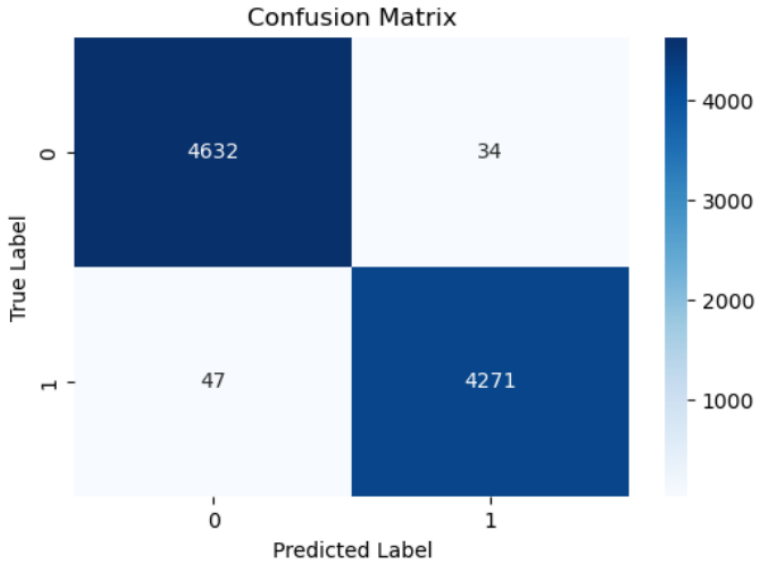

Figure 3. Confusion Matrix

As shown in the figure 3, the confusion matrix plays a key role in evaluating LSTM model accuracy results. It not only provides the overall performance of the model in the task of identifying social media disinformation, but also gives researchers insights into the behaviour of the model in different prediction situations. With the confusion matrix, researchers can calculate a series of important performance metrics such as accuracy, precision, recall and F1 score from the four key metrics (true positives, false negatives, true negatives and false positives). These metrics not only tell people the classification accuracy of the model in general, but also reveal differences in the model's performance on different categories. For example, a high false-negative rate may imply that the model has problems in false information detection, while a high false-positive rate may mean that the model has challenges in real information recognition. Confusion matrices can also help people understand the patterns of model misclassification. For example, by looking at the distribution of false positives and false negatives, people can identify textual features that the model may be prone to confusing, thus providing guidance for further feature engineering or model tuning. In summary, the confusion matrix plays a crucial role in the evaluation of LSTM model accuracy results. It provides a comprehensive performance analysis framework that helps us understand the strengths, weaknesses, and potential room for improvement of the model so that the LSTM model can achieve better performance in identifying social media disinformation.

5. Conclusion

In summary, LSTM shows excellent ability in the identification of social media disinformation. Its deep understanding of complex information patterns, sentence structures, and semantic nuances makes it a reliable guardian. By revealing hidden meanings in posts and comments, LSTM is able to efficiently sift out suspicious content from massive amounts of data. In the chaotic environment of social media, LSTM acts as a trusted ally, ensuring accuracy and revealing the truth about false information. It not only helps to maintain the credibility of information, but also plays an indispensable role in shaping the public's correct perception and building a healthy social media environment.

References

[1] Tacchini E, Ballarin G, Della Vedova ML, Moret S, de Alfaro L (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.

[2] Zhou X, Zafarani R (2019) Network-based fake news detection: a pattern-driven approach. ACM SIGKDD Explor Newsle 21 (2): 48 – 60.

[3] Shu K, Wang S, Liu H (2019) Beyond news contents: the role of social context for fake news detection. In: Proceedings of the twelfth ACM international conference on web search and data mining, pp 312 – 320.

[4] Wang Y, Ma F, Jin Z, Yuan Y, Xun G, Jha K, Su L, Gao J (2018) Eann: Event adversarial neural networks for multi-modal fake news detection. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining, pp 849 – 857.

[5] Feng S, Banerjee R, Choi Y (2012) Syntactic stylometry for deception detection. In: Proceedings of the 50th annual meeting of the association for computational linguistics: short papers, vol 2. Association for Computational Linguistics, pp 171 – 175.

[6] Pérez-Rosas V, Kleinberg B, Lefevre A, Mihalcea R (2018) Automatic detection of fake news. In: Proceedings of the 27th international conference on computational linguistics, pp 3391 – 3401.

[7] Yang Y, Zheng L, Zhang J, Cui Q, Li Z, Yu PS (2018) TI-CNN: convolutional neural networks for fake news detection. arXiv: arXiv - 1806.

[8] Gupta S, Thirukovalluru R, Sinha M, Mannarswamy S (2018) CIMTDetect: a community infused matrix-tensor coupled factorization-based method for fake news detection. In: 2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM). IEEE, pp 278 – 281.

[9] Bisaillon, C. (2019). Fake and real news dataset. Kaggle. https: //www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset.

[10] Irawan, D., Cholissodin, I., Handajani, L., Kholiq, A., & Syuhada, A. (2021). Cytotoxicity effect of ivermectin as an adjuvant treatment in COVID-19 patients. IOP Conference Series: Materials Science and Engineering, 1099 (1), 012040. https: //doi.org/10.1088/1757 - 899X/1099/1/012040.