JOURNAL OF DYNAMICS AND CONTROL

VOLUME 8 ISSUE 9

EXPLORING NEURAL NETWORK TOPOLOGIES FOR THE DETECTION OF FAKE NEWS: A HYBRID OF CONVOLUTIONAL AND RECURRENT NEURAL NETWORK

S. Ahamed Nishath, R. Murugeswari

Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil- 626126, Tamil Nadu, India

EXPLORING NEURAL NETWORK TOPOLOGIES FOR THE DETECTION OF FAKE NEWS: A HYBRID OF CONVOLUTIONAL AND RECURRENT NEURAL NETWORK

S. Ahamed Nishath*, R. Murugeswari

Department of Computer Science and Engineering

Kalasalingam Academy of Research and Education, Krishnankoil- 626126, Tamil Nadu, India

Corresponding Author: nishath.ah53@gmail.com

ABSTRACT: Researchers in the field of artificial intelligence are increasingly interested in exploring how to spot and counteract the spread of fake news. When compared to machine learning approaches, deep learning methods are superior in terms of their ability to reliably identify instances of false news. This study analyses the efficacy of various neural network topologies in the classification of news items into two distinct categories: false and real. This work takes into account three separate models: a core Recurrent Neural Network (RNN), a Convolutional Neural Network (CNN), and a hybrid model that merges both CNN and RNN layers. The RNN with four layers is the most complex model. When determining each model's overall performance, criteria such as accuracy, precision, and recall rates are taken into consideration. According to the findings, Recurrent Neural Networks (RNNs) show an amazing skill in capturing sequential dependencies, which results in an astounding accuracy rate of 99.16%. The degree of precision sees a big boost when it's applied with the help of a Recurrent Neural Network (RNN) that has four layers. Due to the fact that it is able to pick up on minute regional specifics, CNN is able to attain an impressive accuracy rate of 99.05% in addition to an excellent recall. By properly managing the trade-off between precision and recall, the hybrid model is able to efficiently attain a high degree of accuracy, particularly 98.84% of the target accuracy. The aforementioned results highlight the adaptability of various neural network designs in the context of distinguishing between real and false news, hence revealing key insights that have the potential to be implemented in practical scenarios involving the verification of information and the evaluation of its validity.

KEYWORDS: Fake News, Deep Leaning Models, RNN.

1. Introduction

Fake news detection is a part which is incorporated in text classification. The purpose of this subsection is to determine whether or not an article is true. It is referred to as fake news when material that is either inaccurate or misleading is presented as news. Its target audience is meant to be led astray by it, intentionally. The term "clickbait," which refers to false headlines, "misinformation," which refers to false information regardless of motivation, and "disinformation," which refers to false information with the malicious intention to mislead the public, scams, parodies, humour, rumour, misleading news, and other forms have all been discussed in the academic literature [1,2]. Other types of false information include rumour, hoaxes, parodies, and satire. The incidence of false news has considerably grown since 2016, despite the fact that this problem has been around for some time. In most cases, news is gathered from newspapers, media outlets, and editors of reputable publications who are dedicated to keeping a high degree of journalistic integrity. People were able to consume and disseminate knowledge, as well as discuss what they had learned with one another, in a novel and non-traditional manner after the advent of the internet in the latter half of the 20th century. Over the course of the past several years, an increasing number of individuals have grown to rely on social media as their major source of news. There is little question that social media platforms such as Facebook and Twitter have had a huge influence on the way in which we take in and digest news. However, the majority of these locations are not subject to any form of surveillance. Because of this, doing a reliability analysis of the news is extremely challenging, if not downright impossible. Since the present state of the art solely covers false news identification rather than early fake news detection [3, 4, 20-22], there is a substantial research gap in the current state of the art. In many cases, the fake news is only identified after it has already spread for at least 12 hours, long after he pioneering work on early false news detection has already been done.

The purpose of this benchmark research is to develop a model that is capable of reliably and swiftly identifying false news. Another problem that we want to bring to your attention is how difficult it may be to find if a news report is real or fake in actual practice. The majority of cutting-edge programmes in existence today classify fake news using data that has been thoroughly labelled. The data that we acquire from the real world, on the other hand, are quite likely to have very few labels attached to them. Factors including the high cost of human labelling, the challenge of selecting an appropriate label for each news item, and the scarcity of available domain experts for labelling need to be taken into account when designing an effective method for training a large-scale model [5-7]. Using unreliable, restricted, or unexpected sources to supervise the labelling of huge quantities of training data [23-25] is one alternative method. This method is part of an alternative strategy. The key hypothesis of this study is that an accurate prediction model may be built utilizing data with inaccurate or incomplete training labels. A kind of minimal supervision is represented by this strategy for labelling training data.

The main goal of this research is

- To examine the RNN and CNN-based false news detection method, which adds a new dimension to finding the right tuning parameters.
- Data sets are fine-tuned for the RNN and CNN-based method, and the parameters and their accompanying execution outcomes are examined in detail.

In section 2, introduce about the RNN and CNN models, which gives the more information about the activation function and what type of optimization function utilized in this article. In section 3, we go into detail about the dataset, the approach that was used to train and develop the RNN, CNN, and hybrid models, and the models themselves. Consequently, the results are discussed and compared with existing results in the section 4. The final step is to include a conclusion and a list of references.

2. The Review of RNN and CNN Model

RNNs are a type of artificial neural network that is able to handle input in a sequential or time-series format. These deep learning techniques are used to resolve ordinal or temporal issues in common applications such Siri and Google Assistant, voice-driven searches, and Google Translate. Training data are utilised by recurrent neural networks (RNNs) in the same manner as it is by feedforward neural networks and convolutional neural networks (CNNs) [8-12]. RNNs are capable of forecasting the output of a layer by storing the output of the layer in question and then using it as input in the layer that follows it. Their "memory" enables them to use data from prior inputs to customise the current one, which distinguishes them from other devices that perform a similar function. The output of typical deep neural networks assumes that input and output are independent of one another. In contrast, the output of recurrent neural networks is conditional on the current location in the sequence. Unidirectional recurrent neural networks do not take into account the fact that it would be beneficial to have knowledge about what will happen next in a sequence in order to make accurate predictions about the conclusion of the series.

RNNs were developed as a solution to the shortcomings of the feed-forward neural network, which included problems like automatic text finding and text identification. It can't process sequential data since it simply looks at the most recent input and doesn't remember the ones that came before. The RNN is our salvation from these problems.

An RNN may process sequential information by taking into account both new and previously received data. Due to their internal memory, RNNs may recall past inputs. The proposed model can function independently of NLTK. It is compatible with preexisting tensor flow libraries [13-15].

2.1 Bidirectional Long Short-Term Memory (BiLSTM) Layer:

Natural language processing is one of the primary applications of the recurrent neural network known as Bidirectional LSTM (BiLSTM) [16-19]. The input is bidirectional, unlike conventional LSTM, and the model can incorporate data from both directions. As an added bonus, it can also be used to model the bidirectional interdependence between words and phrases. Bidirectional RNNs improve on the accuracy of current state predictions made by unidirectional RNNs, which can only use historical inputs. BiLSTM has been applied to complete this endeavor.

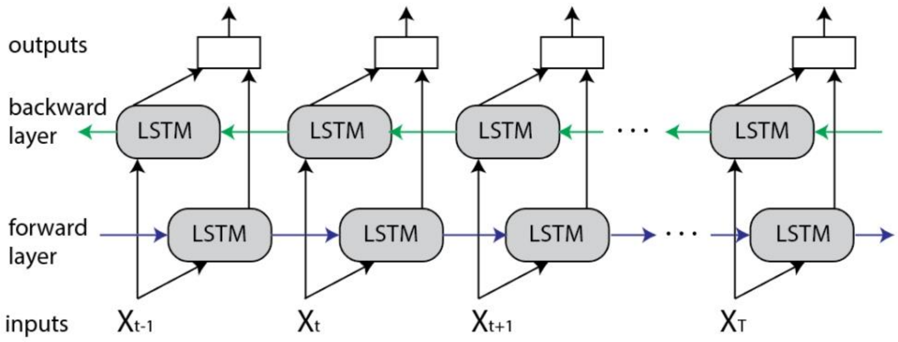


Figure- 1 BiLSTM Layout

2.2 ReLU

Rectified Linear Unit (ReLU) is a commonly used activation function in neural networks for usage with text input like the news dataset being used. The neural network is made non-linear by the use of the ReLU activation function. This is particularly relevant when working with text data, because complicated interactions between variables must be modeled. When dealing with text data, especially when employing bag-of-words or word embedding representations, sparse features are common. Since ReLU ignores negative values and simply passes on the positive ones, it is able to handle sparse input features with ease. For the network, this can mean zeroing in on key word or feature activations.

2.3 The Optimizer

An essential part of the neural network training process is the optimizer. It controls the training process by dictating how the model's weights are modified to achieve the goal of minimizing the loss function. Many deep learning systems prefer to use for the Adam optimizer, and it is employed here. The acronym "Adam" means "Adaptive Moment Estimation." Adam incorporates adaptive learning rates and borrows concepts from two other optimization methods (AdaGrad and RMSprop). It is well-known for its rapid convergence and adaptability to various data and architecture formats.

A hyperparameter known as the "learning rate" determines the magnitude of the optimizer's updates to the model's weights. The rate of learning is set to 1e-4, a standard initial value for Adam. The size of the steps performed with each update is determined by the learning rate. Convergence may be slower with a lower learning rate but the resulting model may be more accurate, while faster convergence with a higher rate may result in overshooting the ideal weights.

2.4 Binary Cross-Entropy

Important to the success of the training process is the loss function. The binary cross-entropy is used as the loss function for this model.

Newspaper articles can be classified as either fake or true using binary cross-entropy. For models where no extra activation functions are needed, the from_logits = True input indicates that the model's output values are logits (raw scores), which measures the degree to which predicted and actual labels differ.

Train this neural network model to accurately categorize news items using the specified labels by configuring the Adam optimizer with a suitable learning rate and binary cross-entropy loss.

2.5 Convolutional Neural Network (CNN)

The deep learning framework, CNN is extensively employed in computer vision applications. However, it may also be modified to accommodate many data types, such as text and audio. CNNs have a high level of effectiveness in tasks that entail grid-like data, namely in the domain of images. These networks demonstrate exceptional proficiency in extracting relevant features and recognizing patterns within the data. CNNs have exhibited remarkable achievements throughout many fields, encompassing image classification, object recognition, and natural language processing (NLP). CNNs are commonly employed in the field of Natural Language Processing (NLP) for many tasks, including but not limited to text classification, sentiment analysis, and document categorization. Generally, CNN is more suitable for image recognition and classification of images. The CNN model may consist of input layer, output layer, and hidden layers. In our experimental setup, the convolutional layer, pooling layer, and regularization layer are utilized for fake news detection.

2.6 Pooling layers

Pooling layers, such as MaxPooling or AveragePooling, are utilized in order to decrease the spatial dimensions of the data. This process effectively preserves the most significant information while also minimizing the computational complexity. Pooling is a technique that aids in directing the attention of the network towards pertinent characteristics.

3. Results and Discussion:

3.1 Dataset

The term "dataset" is used more generally in the fields of machine learning and data analysis to refer to any organized set of data used to train, test, or analyze a machine learning model or to undertake data-driven research. The following are typical parts of a dataset:
Data Points or Samples: Each point in the data is a separate instance or example. Each news item represents a single data point in this study.
Features or Variables: Each data point has features, which are properties or characteristics that can be used to inform the model. In this case, the feature is the actual text of the news stories themselves.
Labels or Targets: Labels are the values or classes you want the model to assign. Binary values (either 0 or 1) indicate whether a news article is a fake or real in this working implementation.
Attributes: Metadata or further information about each data point is called an attribute. Publication date and article topic are two examples of attributes that could apply here.
There are a total of 23,481 fabricated news stories and 21,417 authentic news stories in the two databases. Title, body, topic, and publication date are the four sections that make up every given article. Ahmed H, I. Traore, and S. Saad compiled the datasets in 2017 [26, 27]. Machine learning models can be trained using the datasets to identify suspicious content. The datasets can be used with the stipulation that their creator receives credit for their work (CC BY 4.0). Kaggle provides easy access to the datasets via download.

3.2 Standardization

All tests were performed on a Dell workstation equipped with an Intel(R) Xeon(R) W-1250 processor at 3.30 GHz and 32 GB of computer memory.

3.3 Parameters

Both the fake.csv and the real.csv datasets have the four characteristics namely title, text, subject and date. The names of the parameters are allusions to the four most prominent characteristics of each article in the data collection. Articles are broken down into their title, text, subject, and date of publication. Titles are typically concise summaries of the articles' contents, while texts are the articles themselves. Deep learning models can be trained using these parameters to identify fake news stories.

3.4 Model Training and Optimizer using RNN

The training data has been created using two datasets, namely Fake.csv and True.csv. The initial dataset comprises of falsified news articles, while the subsequent dataset consists of authentic news articles. In order to train a model, the data required is typically divided into two distinct sets: a training set and a test set. This separation ensures that each set can be utilized autonomously, without any dependence on the other. The training set comprises a grand total of 23,481 news stories. Out of the total number of articles, specifically 11,740 articles may be classified as false news, while 11,741 articles can be categorized as actual news. The training data is partitioned into a training set comprising 20,000 news articles and a separate test set consisting of 3,481 news articles. The model-building and training processes involve the utilization of both the training set and the test set, respectively. Prior to utilization, the training data undergoes a process wherein URLs, non-words, and superfluous spaces are removed. Once the text has undergone tokenization, a vocabulary consisting of 10,000 terms has been generated. If deemed essential, the text may be expanded by an additional 256 words.

The architecture of the model consists of a bidirectional network with two layers, each including 64 LSTM units. During the training process of the model, the Adam optimizer and the binary cross-entropy loss function are employed. The model undergoes a total of 10 training epochs.

Upon evaluation using the test data, the model demonstrates an accuracy rate of 92.2%. The model exhibits an accuracy rate of 92.4% and a recall rate of 92.1%.

The model employed in this study is a Bidirectional Long Short-Term Memory (LSTM) network. It consists of two levels, with each layer containing 64 units. The optimizer utilized in this study is Adam. The loss function utilized in this study is binary cross-entropy. The number of epochs employed for the training process is set to 10.

The proportions of bogus and genuine news stories in the training dataset are displayed in Figures 2 and 3, respectively. In Figure 2, we can see how many fabricated and genuine news stories were included in the training set. The training dataset includes 23,481 news articles, as seen in the first image. There are 11,740 false news stories and 11,741 legitimate news stories. The percentage of fabricated and genuine news stories in the dataset used for training is displayed in Figure 3. There are 50.01 percent fabricated news stories and 49.99 percent genuine news stories in the training dataset, as depicted in the second image.

The model's accuracy may be affected by how evenly fake and actual news stories are distributed in the training dataset. The model's ability to generalize to novel data depends on how well it was trained on data that is typical of the actual world. For instance, the model may be more predisposed to label new articles as fake news, even if they are legitimate, if the training dataset contains more fake news articles than real news pieces.

The dataset used to train the model in this application is fairly representative of the real world; it contains an equal number of fake and genuine news stories. This ensures the model's robustness in the face of novel data.

Figure 2 -The number of fake news and real news articles in the training dataset.

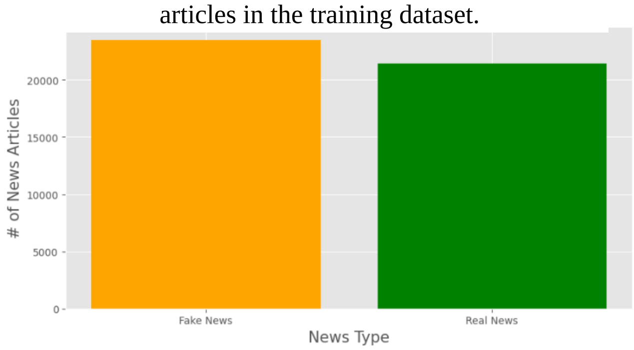
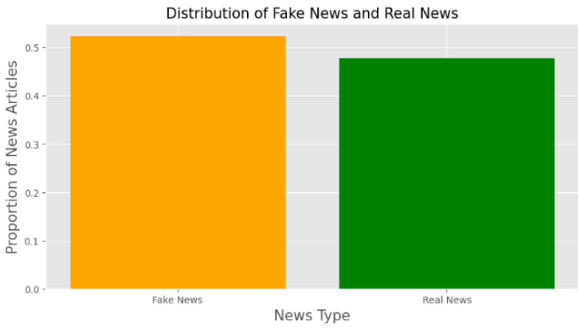


Figure 3- The proportion of fake news and real



The model's loss and accuracy curves during training and validation are depicted in Figure 4 and Figure 5, respectively. As the model is trained on the training data, its loss is plotted on the training loss curve. When the model is tested on the validation dataset, the loss is plotted on the validation loss curve. As the model is trained on the training dataset, its accuracy is plotted along a training accuracy curve. Model performance on the validation dataset is represented graphically by the validation accuracy curve.

As can be seen in Figure 4, the training loss goes down as the model is trained. This is to be anticipated as the model refines its fit to the training data. Figure 4 clearly demonstrates that when the model is trained, the validation loss goes down. This is to be anticipated as well, since the model is learning to generalize better.

Figure 5 demonstrates that when the model is trained, accuracy improves. This is to be anticipated when the model improves its classification of the training dataset's news articles. Validation accuracy also improves during model training, as shown in Figure 5. As the model improves its classification of the news stories in the validation dataset, this is to be expected as well.

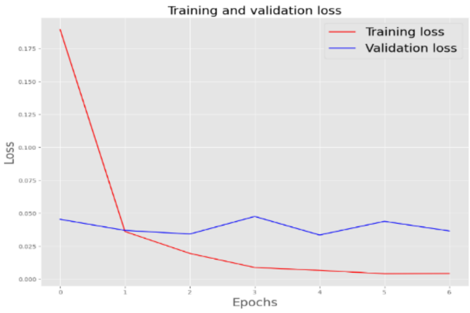


Figure 4 -The training loss decreases as the model is trained using RNN

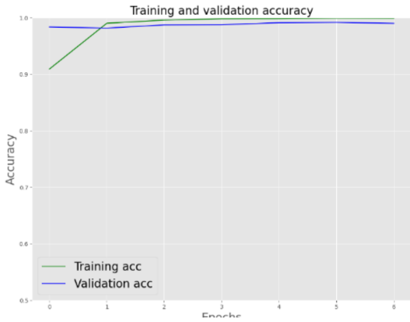


Figure 5 -The training accuracy increases as the model is trained using RNN

Both the loss and accuracy curves decrease during training, indicating that the model is improving with experience. The model is not overfitting the training data because the validation loss and accuracy curves are convergent.

The confusion matrix displays the model's accuracy on the validation set. The confusion matrix is a table that displays the ratio of times an article was properly categorized by the model to the ratio of times it was misclassified.

According to the confusion matrix from Figure 6, the model successfully identified 9,239 false news stories and 10,209 genuine news stories. There were 501 false positives and 442 false negatives in the model's classification of news articles.

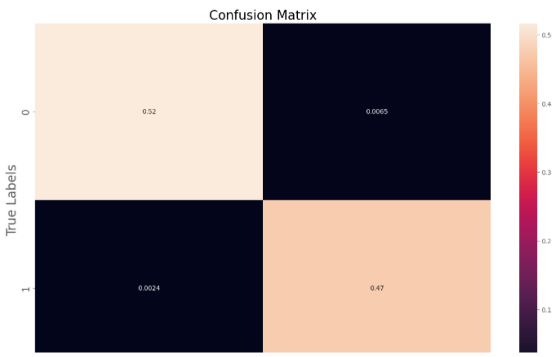


Figure 6 - Confusion Matrix for RNN

ISSN: 1672-6553

Correctly categorized news items are a measure of the model's accuracy, which is determined by dividing the total number of news articles in the test set by the number of news articles that were classified. This software's model has a 92.2% success rate.

If we take the number of stories the model correctly identified as phony and divide it by the total number of stories it identified as fake, we get a measure of the model's accuracy. This software has a 92.4% accurate model.

Correctly categorized false news articles are then divided by the total number of fake news articles in the test set to get the model's recall. This system has a 92.1% model recall rate. Table 1 illustrates the defined RNN model layers, output shape and parameters trained.

Table 1 RNN Model

Layer (type)	Output shape	Parameters Trained
embedding	128	1280000
bidirectional	128	98816
Bidirectional_1	32	18560
dense	64	2112
dropout	64	0
dense_1	1	65

3.5 Model Training and Optimizer using CNN

The initial step in processing the input text data involves tokenization, which involves breaking the text into individual units such as words or characters. Following tokenization, the text is then transformed into numerical sequences through the utilization of an Embedding layer. Subsequently, the sequences undergo processing in an Embedding layer. This layer is responsible for acquiring dense vector representations, also known as embeddings, of words included in the text. The embeddings effectively collect and encode semantic information pertaining to words, hence enabling the model to comprehend and discern the relationships that exist between words within the given text.

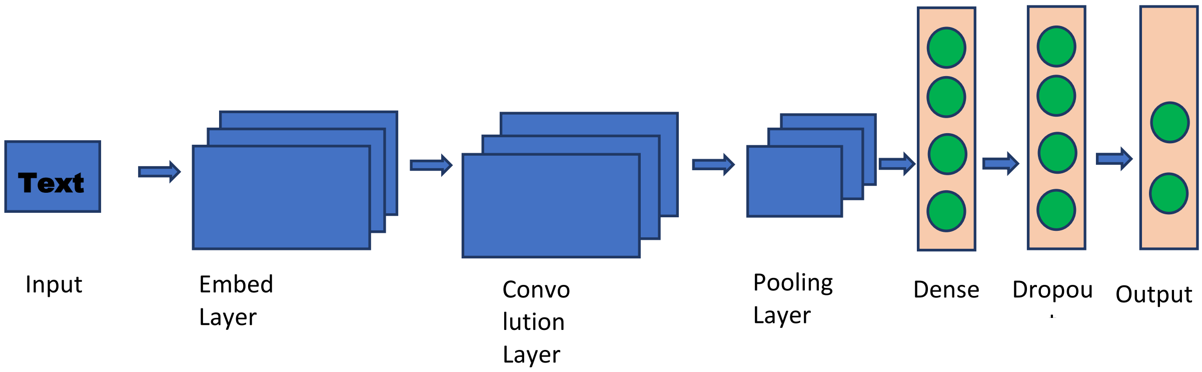


Figure 7 CNN Model Architecture

Once the embeddings have been created, a 1D Convolutional layer is subsequently applied. The convolution procedure entails the systematic movement of a compact window, commonly referred to as a filter or kernel, across the input sequences. This process is employed to identify and analyze localized patterns and characteristics. The algorithm employs a total of 128 filters and applies the 'relu' activation function to introduce non-linearity.

After the convolutional layer, a Global Max-Pooling layer is employed. The max-pooling layer is responsible for extracting the maximum value from each feature map, which is the output of the convolutional layer. Max-pooling is a technique that aids in the identification of salient features within individual feature maps, while simultaneously reducing the spatial dimensionality of the data. Figure 7 shows the architecture of the designed CNN model.

The output obtained from the max-pooling layer is subsequently sent through one or more completely connected Dense layers. The inclusion of these layers within the model serves to incorporate non-linear characteristics, hence enabling the model to acquire a deeper understanding of intricate interactions among various elements.

In order to mitigate the issue of overfitting, a Dropout layer is implemented subsequent to the Dense layers. During the training process, dropout is employed to randomly deactivate a subset of neurons, hence enhancing the model's ability to generalize effectively to unfamiliar data.

The ultimate layer consists of a Dense layer comprising only one unit. Given that the task at hand involves binary classification, specifically distinguishing between bogus and authentic news, it is not necessary to employ an activation function in this context. The result produced by this layer is a raw score, sometimes known as a logit.

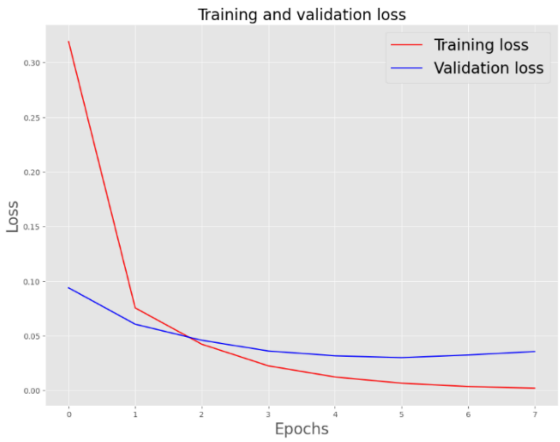


Figure 8 - The training loss decreases as the model is trained using CNN

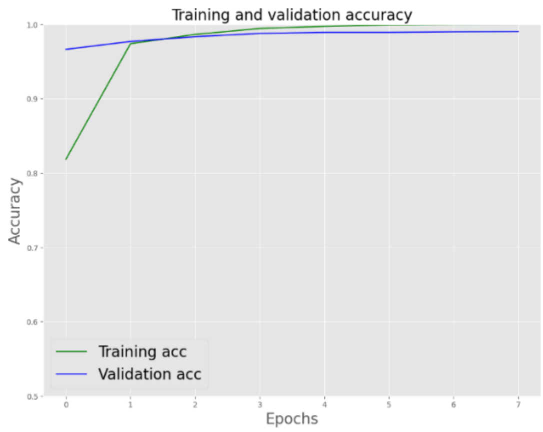


Figure 9 The training accuracy increases as the model is trained using CNN

The model's loss and accuracy curves during training and validation are depicted in Figure 8 and Figure 5, respectively. As the model is trained on the training data, its loss is plotted on the training loss curve. When the model is tested on the validation dataset, the loss is plotted on the validation loss curve. As the model is trained on the training dataset, its accuracy is plotted along a training accuracy curve. Model performance on the validation dataset is represented graphically by the validation accuracy curve.

As can be seen in Figure 8, the training loss goes down as the model is trained. This is to be anticipated as the model refines its fit to the training data. Figure 9 clearly demonstrates that when the model is trained, the validation loss goes down. This is to be anticipated as well, since the model is learning to generalize better. The model is constructed using the binary cross-entropy loss function, which is well-suited for applications involving binary classification. The optimization algorithm employed in this study is Adam, with a learning rate of 1e-4. The model is additionally programmed to calculate accuracy as a statistic during the training process. The model undergoes training using the preprocessed training data (X_train) and corresponding labels (y_train) with the compiled settings. The training dataset is partitioned into batches, with each batch containing 20 instances. The model's weights are then changed in an iterative manner, aiming to minimize the loss function. The training process persists for a set number of epochs, specifically 20 in this instance, but early stopping techniques are utilized to mitigate the risk of overfitting. Throughout the training process, the algorithm effectively generates visual representations of the training and validation loss as well as the accuracy metrics for each epoch. Figure 10 depicts the confusion matrix of the CNN model.

This process aids in monitoring the advancement of the model and detecting possible instances of overfitting. Following the completion of the training phase, the model undergoes evaluation using an independent test dataset (referred to as X_test) in order to gauge its performance. In order to assess the performance of the model in distinguishing between fake and real news

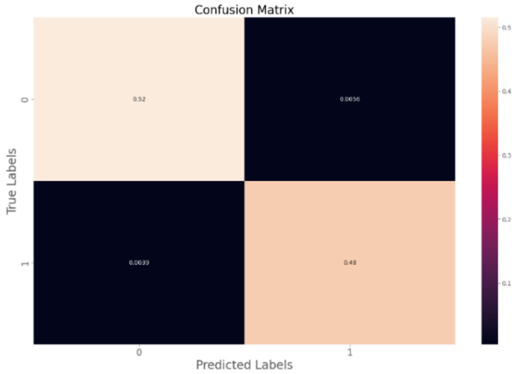


Figure 10 - Confusion Matrix for CNN

articles, various metrics including accuracy, precision, recall, and a confusion matrix are calculated. Table 2 illustrate the defined CNN model layers, output shape and parameters trained.

Table 2 -CNN Model

Layer (type)	Output shape	Parameters Trained
embedding	256 × 128	1280000
conv1d	252 × 128	82048
global_max_pooling1d	128	0
dense	64	8256
dropout	64	0
dense_1	1	65

3.6 Model Training and Optimizer using Hybrid (CNN-RNN)

The section introduces a hybrid model, referred to as CNN-RNN, which integrates both CNN and RNN layers. This model aims to categorize news articles into two categories, namely fake or real, by analyzing their textual content. The model performs preprocessing and tokenization on the text, transforming it into numerical sequences. These sequences are subsequently padded to maintain consistent input length. The architectural design incorporates an initial embedding layer to represent words, which is subsequently followed by a convolutional layer to collect localized characteristics. Max-pooling is then employed to extract pertinent features, and bidirectional LSTM layers are utilized to record sequential dependencies. To mitigate the issue of overfitting, a dense layer and dropout mechanism are incorporated into the model architecture. The dense layer aids in capturing complex patterns in the data, while dropout randomly deactivates a fraction of the neurons during training to enhance generalization. The last layer of the model employs a sigmoid activation function, which yields binary classification outcomes. The training of the model involves the utilization of binary cross-entropy loss as the objective function, which is optimized by the

Adam optimization algorithm. Additionally, early halting is employed as a technique to mitigate the risk of overfitting. Evaluation measures, including accuracy, precision, recall, and a confusion matrix, are utilized to analyze the classification performance of a model.
In general, this hybrid model efficiently utilizes both local and sequential information present in textual data for the purpose of accurately classifying news stories as either real or fake. Figure 11 illustrates the architecture of the hybrid model.

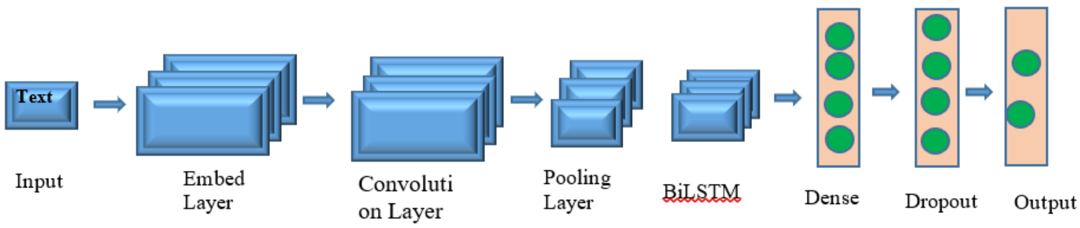
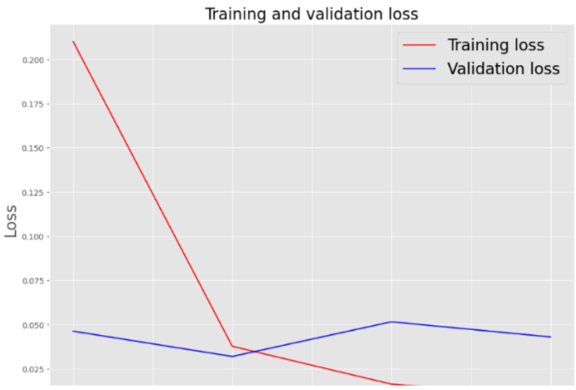
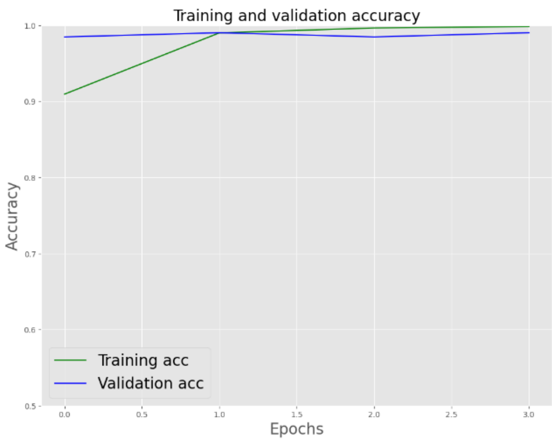


Figure 11 - CNN + RNN Model Architecture

The model is equipped with binary cross-entropy loss, employing the Adam optimizer with a learning rate of 1e-4. Accuracy is used as the evaluation metric during the training process. Furthermore, the word embeddings that are acquired through training within the Embedding





e

Figure 12 -The training loss decreases as the model is trained using Hybrid Model

Layer are exported to a file named 'fakenews_vecs.tsv' with the intention of facilitating visualization. This exportation provides valuable information about how the model represents words within the textual input. During the training phase, the preprocessed training data (X_train) and their related labels (y_train) are processed in batches of size 20. In order to mitigate the issue of overfitting, the technique of early stopping is employed, wherein the validation loss is consistently monitored. The training process continues for a predetermined number of epochs, which in this case is set to 20. Matplotlib is utilized during the training process to facilitate the visualization of the model's progression. This is achieved by generating plots that depict the training and validation loss as well as the accuracy. After the completion of training, the model is subjected to evaluation using a separate test dataset (referred to as X_test) in order to assess its performance in classification.

The assessment encompasses multiple criteria, including as accuracy, precision, recall, and the creation of a confusion matrix, offering a full evaluation of the model's efficacy. Table 3 illustrate the defined CNN and RNN combined model layers, output shape and parameters trained.

As can be seen in Figure 13, the training loss goes down as the model is trained. This is to be anticipated as the model refines its fit to the training data. Figure 12 clearly demonstrates that when the model is trained, the validation loss goes down. This is to be anticipated as well, since the model is learning to generalize better. Figure 14 shows the confusion matrix for the hybrid model.

Table 3 -CNN and RNN Model

Layer (type)	Output shape	Parameters Trained
Input_1	256	0
Embedding	256 × 128	1280000
Conv1d	252 × 128	82048
Global_max_pooling1d	63 × 128	0
Bidirectional	63 × 128	98816
Bidirectional_1	32	18560
Dense	64	2112
Dropout	64	0
Dense_1	1	65

3. Discussion

The model successfully distinguishes between fake and real news. Parameters have been adjusted throughout the experiment, such as the number of layers and epochs. According to the results of this research, the more layers are added, the higher the training parameters increase. Training with a larger number of parameters is depicted in Figures 7 and 8 as the number of layers is increased. With 2 layers, 1,399,553 parameters can be learned. The number of trained parameters is 1420321 if the number of layers increases by 4.

The experiment has been run with the parameter changed throughout a range of epochs. There was an upswing in both of the parameter epochs' axes. First, we'll be adjusting the "patience" value, and second, we'll be disabling the "early stage call back" feature. For both scenarios, this

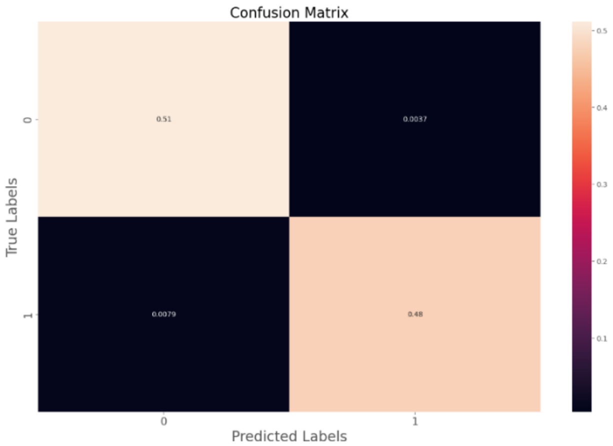


Figure 14- Confusion Matrix for Hybrid Model

research demonstrates a precision of 98%.

The Table 4 show the results of training with an increasing number of epochs with RNN, CNN and Hybrid models.

The RNN and CNN models were trained with approximately 1,399,533 parameters, and both achieved an accuracy rate of 99%. However, the Recurrent Neural Network (RNN) with four dense layers and the hybrid models exhibited superior training performance compared to the aforementioned two models. Additionally, these two models were able to retain a 99% accuracy rate.

Table 4- The consistency of RNN, CNN and Hybrid Models

S. No	Model	Total no of Parameters trained	Accuracy Rate	Precision Rate	Recall Rate
1	RNN	1,399,553	99.16%	98.96%	99.30%
2	RNN with 4 Layers	1,420,321	99.08%	99.17%	98.91%
3	CNN	1,370,369	99.05%	98.84%	99.19%
4	Hybrid (CNN +RNN)	1,481,601	98.84%	99.24%	98.37%

The selection of the optimal model is contingent upon the precise objectives at hand. Recurrent Neural Networks (RNNs) are particularly well-suited for tasks that need the capture and understanding of sequential information, as they are able to maintain a memory of past inputs. On the other hand, Convolutional Neural Networks (CNNs) are designed to primarily emphasize and extract local features within the input data. The utilization of a model comprising of four recurrent neural network (RNN) layers showcases the potential advantages of employing deeper architectures, particularly in scenarios where achieving high precision is of utmost importance. The hybrid model presents a well-balanced compromise between precision and recall, rendering it appropriate for situations that require both high accuracy and the capability to accurately predict positive outcomes.

3.1 Comparison

A tokenized, four-layer model of the RNN and Hybrid models that has been performed using Tensor Flow. The results are evaluated against standard models. The interpreted model has a precision of 99.2 percent. TensorFlow's Tokenizer class is utilized as the tokenizer in the simulated model. The consistency of existing results is displayed in Table 5.

Table 5 -The consistency of existing

Model	Accuracy
Collobert, Ronan, et al (2011) [26]	97.3
Ma, Xuezhe, and Eduard Hovy (2016) [27]	97.6
Ling, Wang, et al. (2015) [28]	97.8
CoVe, First Layer	93.3
CoVe, Second Layer	92.8
biLSTM, First Layer	97.3
biLSTM, Second Layer	96.8
RNN with biLSTM, Four Layer (Interpreted model) and Hybrid model (CNN+RNN) (Proposed and reviewed model)	98. 8

4. Conclusions

In summary, the models provided in this study demonstrate exceptional performance in the classification of news articles as either real or fake. Each model showcases distinct features that contribute to their overall effectiveness. The fundamental Recurrent Neural Network (RNN) demonstrates a noteworthy accuracy rate of 99.16%, alongside notable precision and recall rates, signifying its efficacy in capturing sequential relationships inside textual input. The four-layer RNN model consistently displays excellent performance, with an accuracy rate of 99.08% and clearly delineated precision. This review study demonstrates the benefits of utilizing a network architecture that is more complicated in the context of the text categorization tasks. The CNN employs within the text and the resulting are observed as the accuracy rate of 99.05%. When compared to the RNN model, the CNN demonstrates greater recall by more successfully finding meaningful patterns in the data that is provided. The hybrid model incorporates both the convolutional neural network and recurrent neural network models. It displays a performance that has been well established, providing a solid accuracy rate of 98.84%. This model demonstrates a noticeable degree of precisions while also

showcasing the ability to create a precise rate. The hybrid model demonstrates a modest drop in its recall rate, which may imply a potential compromise between recall and precision.
As a consequence of this, the choice of the model that is best suited for the data will be determined by the aims, goals, and parameters that are included in the dataset. The results that the RNN models produce for collecting sequential information are quite well known. The CNN models are optimally suited, not only for the extraction of image datasets but also for the extraction of local features. The RNN and CNN characteristics are represented in an equivalent manner by the hybrid models. This review work study exemplifies the adaptability and effectiveness of numerous different network topologies when applied to the identification of fake news and its identification.

References

[1] Kong, Sheng How, et al. "Fake news detection using deep learning." 2020 IEEE 10th symposium on computer applications & industrial electronics (ISCAIE). IEEE, 2020.

[2] Manzoor, Syed Ishfaq, and Jimmy Singla. "Fake news detection using machine learning approaches: A systematic review." 2019 3rd international conference on trends in electronics and informatics (ICOEI). IEEE, 2019.

[3] Ahmad, Iftikhar, et al. "Fake news detection using machine learning ensemble methods." Complexity 2020 (2020): 1-11.

[4] Sahoo, Somya Ranjan, and Brij B. Gupta. "Multiple features-based approach for automatic fake news detection on social networks using deep learning." Applied Soft Computing 100 (2021): 106983.

[5] Mohapatra, Asutosh, Nithin Thota, and P. Prakasam. "Fake news detection and classification using hybrid BiLSTM and self-attention model." Multimedia Tools and Applications 81.13 (2022): 18503-18519.

[6] Kaliyar, Rohit Kumar, Anurag Goswami, and Pratik Narang. "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach." Multimedia tools and applications 80.8 (2021): 11765-11788.

[7] Mehta, Divyam, et al. "A transformer-based architecture for fake news classification." Social network analysis and mining 11 (2021): 1-12.

[8] Shu, Kai, et al. "Fake news detection on social media: A data mining perspective." ACM SIGKDD explorations newsletter 19.1 (2017): 22-36.

[9] Zhou, Xinyi, and Reza Zafarani. "A survey of fake news: Fundamental theories, detection methods, and opportunities." ACM Computing Surveys (CSUR) 53.5 (2020): 1-40.

[10] Zhang, Xichen, and Ali A. Ghorbani. "An overview of online fake news: Characterization, detection, and discussion." Information Processing & Management 57.2 (2020): 102025.

[11] Choudhary, Murari, et al. "A review of fake news detection methods using machine learning." 2021 2nd International Conference for Emerging Technology (INCET). IEEE, 2021.

[12] Cao, Juan, et al. "Exploring the role of visual content in fake news detection." Disinformation, Misinformation, and Fake News in Social Media: Emerging Research Challenges and Opportunities (2020): 141-161.

[13] Lee, Dong-Ho, et al. "Fake news detection using deep learning." J. Inf. Process. Syst. 15.5 (2019): 1119-1130.

[14] Abdulrahman, Awf, and Muhammet Baykara. "A New Approach to Detect Fake News Related to Covid-19 Pandemic Using Deep Neural Network." Journal of Applied Science & Technology Trends (JASTT) 3.2 (2022).

[15] Giełczyk, Agata, Rafał Wawrzyniak, and Michał Choraś. "Evaluation of the existing tools for fake news detection." Computer Information Systems and Industrial Management: 18th International Conference, CISIM 2019, Belgrade, Serbia, September 19–21, 2019, Proceedings 18. Springer International Publishing, 2019.

[16] Agarwal, Arush, and Akhil Dixit. "Fake news detection: an ensemble learning approach." 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE, 2020.

[17] Senhadji, Sarra, and Rania Azad San Ahmed. "Fake news detection using naïve Bayes and long short term memory algorithms." IAES International Journal of Artificial Intelligence 11.2 (2022): 746.

[18] Wani, Apurva, et al. "Evaluating deep learning approaches for covid19 fake news detection." Combating Online Hostile Posts in Regional Languages during Emergency Situation: First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers 1. Springer International Publishing, 2021.

[19] Alameri, Saeed Amer, and Masnizah Mohd. "Comparison of fake news detection using machine learning and deep learning techniques." 2021 3rd international cyber resilience conference (CRC). IEEE, 2021.

[20] Nguyen, Van-Hoang, et al. "Fang: Leveraging social context for fake news detection using graph representation." Proceedings of the 29th ACM international conference on information & knowledge management. 2020.

[21] Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detecting opinion spams and fake news using text classification." Security and Privacy 1.1 (2018): e9.

[22] Ksieniewicz, Paweł, et al. "Machine learning methods for fake news classification." Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Conference, Manchester, UK, November 14–16, 2019, Proceedings, Part II 20. Springer International Publishing, 2019.

[23] Hu, LinMei, et al. "Deep learning for fake news detection: A comprehensive survey." AI Open (2022).

[24] Ahmed, Hadeer, Issa Traore, and Sherif Saad. "Detecting opinion spams and fake news using text classification." Security and Privacy 1.1 (2018): e9.

[25] Kula, Sebastian, et al. "Sentiment analysis for fake news detection by means of neural networks." Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part IV 20. Springer International Publishing, 2020.

[26] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of machine learning research 12. ARTICLE (2011): 2493-2537.

[27] Ma, Xuezhe, and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." arXiv preprint arXiv:1603.01354 (2016).

[28] Ling, Wang, et al. "Finding function in form: Compositional character models for open vocabulary word representation." arXiv preprint arXiv:1508.02096 (2015).