Fake news classification using machine learning and deep learning techniques

Pieter van Brakel
Student number: 2058059

Thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Data Science & Society
Department of Cognitive Science & Artificial Intelligence
School of Humanities and Digital Sciences
Tilburg University

Thesis committee:

Dr. Drew Hendrickson
Dr. Silvy Collin

Wordcount: 7928
Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
January 2023

Contents

Preface

Dear reader,

Thank you for taking the time and effort to read this thesis. Before this paper will go into the research itself, I would like to express my gratitude to some people who have guided me in the process of writing the thesis. First, to dr. Drew Hendrickson, who has helped me during the whole process as supervisor. Next, to my fellow students who have helped me make progress with feedback at various moments. Finally, my family and friends, who have been an encouraging source of support.

Sincerely,

Pieter van Brakel

Abstract

Fake news contains false information and can cause harm. Research shows that humans are vulnerable to fake news because of an inadequacy to separate true from fake news. Automatic fake news detection can help in separating true news from fake news. This research compares promising models from the scientific literature to classify fake news articles using news content. Several machine learning and deep learning algorithms will be tested on the WELFake dataset, introduced in (Verma et al., 2021). The Bi-directional RNN-LSTM achieves the highest classification accuracy of 97.19%, higher than the state-of-the-art classifier. Before a fake news detection classifier would be implemented on a larger scale, further research is required.

Ethics Statement

Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data that was used in this thesis retains ownership of the data during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data. The author of this thesis has evaluated his project according to the "Ethics checklist Student research with human participants." The code used in this thesis is not publicly available. Images used in this thesis, when not produced by the author, were licensed under Creative Commons.

Fake news classification using machine learning and deep learning techniques

Pieter van Brakel

January 13, 2023

1 Introduction

News consumption has for a substantial part shifted from offline to online. This shift was visible in the research of (Walker & Matsa, 2021), who found that 48% of Americans consume their news sometimes or often from social media. When news is consumed on social media, its consumers can have a hard time separating truth from falsities. This difficulty is illustrated in the research of (Moravec et al., 2018), whose conclusion was that 56% of Facebook users can not recognize false information that is in alignment with their confirmation bias. Without consumers being able enough to separate truth from falsities within online news, fake news has an opportunity to spread more easily. This spread of fake news provides news consumers with false information and has the potential to cause harm (Giachanou & Rosso, 2020).

To tackle the spread of fake news, multiple strategies have been identified. The article of (Bergstrom & West, 2021) identified technological advancement as one of these strategies. Data science research has attempted to contribute to the technological advancement strategy by developing fake news detection models. The goal of fake news detection research is generally to optimize the classification accuracy of classification models. This optimization is done by comparing classification accuracies using different datasets, features, and algorithms.

This research adopts the goal to improve the current highest classification accuracy. This research adds to the scientific literature by conducting experiments that have not been done on this specific dataset but have achieved high classification accuracies on other datasets. In doing this, this research adds information to the literature about the effectiveness of promising approaches on the task of fake news detection. The aim to increase the highest classification accuracy helps to prevent misclassifications. By preventing misclassifications, potential damage to the trust in these models will be minimized. As trust might be a crucial requirement for a constructive impact on society, preventing a reduction of trust is of societal benefit.

The dataset that will be used is the WELFake dataset that was presented by (Verma et al., 2021). On the WELFake dataset, several machine learning models have been tested. The model that achieved the highest classification accuracy using news content as feature was a Support Vector Machine using frequency-based word representations extracted by bag-of-words. This achieved a classification accuracy of 95.61%. This model will be reproduced as the baseline model for this research and compared to alternatives.

The comparison will be done with promising models from previous research. The classification accuracies these models achieved on other datasets will be addressed in the related works. None of these other models have been tested on the WELFake dataset, which is how this research adds its value. First, the baseline model will be compared with the Support Vector Machine using vector-based word representations extracted by Word2Vec. Then, it will be compared with the Linear Support Vector Machine using frequency-based word representations extracted by bag-of-words. The last algorithm that will be tested is a Bi-directional Recurrent Neural Network with Long Short-Term Memory. The input that will be used for the model is news content.

Lastly, a confusion matrix will be used to gather insight into asymmetries or biases. This confusion matrix will be applied on the model with the highest classification accuracy and it will be split by sub-dataset.

This leads to the following research question and sub-questions:

Research question: How accurately can news articles be classified as fake news by their content using machine learning and deep learning techniques?

Sub question 1: to what extent do a Support Vector Machine with vector-based word representation extracted by Word2Vec, a Linear Support Vector Machine with frequency-based word representation extracted by bag-of-words and a Bi-directional Recurrent Neural Network with Long Short-Term Memory improve classification accuracy over a Support Vector Machine with frequency-based word representation extracted by bag of words?

Sub question 2: to what extent are there asymmetries/biases in the conclusion from the confusion matrix in the model that performs with the highest classification accuracy split by sub-dataset?

As the results will show, the Bi-directional RNN-LSTM achieves the highest classification accuracy with 97.19%. It is an improvement into the the state-of-the-art classifier because it achieves a higher classification accuracy than the baseline model. In answering sub question 2, the results show a bias in the neural network towards classifying news articles as true. The findings will be explained in more detail later in this thesis.

2 Related Work

This research attempts to achieve the highest classification accuracy in the binary classification task of fake news detection. This related work section will dive into the existing literature concerning the topic. It will start of by discussing the literature on fake news in general. Then, it will move on to the data science literature on fake news detection in previous research. It will end with the datasets that have been used in previous research and the dataset that will be used for this research.

2.1 Fake News

This section starts with the definition of fake news. According to (Egelhofer & Lecheler, 2019), fake news is defined as news with a low degree of facticity, an intent to deceive and spread through a journalistic format. (Bergstrom & West, 2021) distinguishes fake news from disinformation or misinformation. Disinformation is the more general category of knowingly spreading false information. Fake news is a subcategory of disinformation. Disinformation is distinguished from misinformation by analyzing the intent of spreading the information. Where disinformation is intended to spread falsities, misinformation spreads false information without the intent to spread falsities. This research focusses specifically on fake news, not disinformation or misinformation.

(Di Domenico et al., 2021) states that fake news spreads through either humans or non-humans with the goal of seeking a response from the individuals that consume fake news. In the case of humans, several experimental studies have tested predictors of individuals believing fake news. The summarizing article of (Bryanov & Vziatysheva, 2021) categorized these predictors into one of three clusters: message characteristics, individual susceptibility to fake news and the accuracy-promoting interventions. These three predictors will be discussed here.

First, the message characteristics. (Vosoughi et al., 2018) identifies the topic as one of these characteristics: fake news messages are shared more when the topic is novel and when it concerns politics. (Lazer et al., 2018) names the medium on which the messages are shared as another characteristic: fake news stories tend to go viral specifically on social media. The research of (Mourão & Robertson, 2019) found that fake news messages tend to be written with only a moderate amount of sensationalism, misinformation, and partisanship. Completely fabricated fake news messages were uncommon. Finally, (Grinberg et al., 2019) indicates that the message affects only tiny fractions of the population. In the 2016 United States presidential election, only 1% of individuals accounted for 80% of fake news source exposure.

Next, the literature on the individual susceptibility. A popular idea is that individual vulnerability to fake news is caused by political polarization, as humans favor news as true when it is in alignment with their political convictions. The article of (Pennycook & Rand, 2021) rejects this idea, because the data in the article shows that people achieve higher classification accuracy when judging news that is in alignment with their political preference compared to news that is not. A different idea was presented in the research of (?, ?). This research showed that individuals with lower levels of cognitive abilities are more susceptible to fake news and are also less likely to change their attitudes after they discover that their previous conviction was based on fake news. A third article from (Pennycook & Rand, 2020) suggests the individuals mindset as a predictor. Individuals who are considered reflexive open-minded tend to be less able to differentiate between fake and real news, since data shows these individuals to be overly accepting of weak claims.

The last predictor discussed is the accuracy-promoting interventions. (Bryanov & Vziatysheva, 2021) summarizes the literature on fake news and distinguishes two major accuracy-promoting approaches in the fight against fake news. The first approach is alerting individuals on the possibility of online deception and equipping them with tools to combat it. The other approach is labeling questionable news stories or sources. Experimental psychological research from (Pennycook et al., 2020) already confirmed the effectiveness of the alerting strategy. The effect of labeling has also been tested in (Brashier et al., 2021) and shown to be effective, especially when timed right. Labeling has been most effective after exposure to the news article, compared to during exposure or before exposure. A drawback on the labeling approach was mentioned by (Tandoc Jr, 2019). This article considered it could backfire and instead increase belief in fake news. The labeling approach is exercised by data science research on the topic of fake news detection.

2.2 Fake News Detection

This section will discuss information about fake news detection from the scientific literature. According to (Verma et al., 2022), three criteria are identified in the literature that help categorize fake news effectively: news propagation (the spreading pattern), user profile (individuals' behavior and information) and news content. News content criteria are writing patterns, such as the number of special characters or verbs, and the representation and structure of the text. This research focuses on improving fake news detection using news content. This focus is chosen because of the specific gaps in the current literature in this area and the availability of a dataset that is suitable for this task. These gaps in the literature will be addressed in this section. First, the experiments using machine learning algorithms and their word embeddings will be discussed. The second part will go deeper into the experiments that use neural network architectures.

2.2.1 Machine Learning Classifiers

(Ahmed et al., 2017) compared each possible combination of two different word embeddings and six different machine learning classifiers. The two-word embeddings were Term Frequency and Term Frequency-Inverted Document Frequency (now called TF-IDF). The six different machine learning classifiers that were tested were Linear Regression (LR), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), Linear Support Vector Machine (LSVM), K-Nearest Neighbor (K-NN) and Decision Tree (DT). The highest classification accuracy obtained was 92% LSVM using TF-IDF LSVM. The authors concluded that linear-based classifiers achieved better results than nonlinear ones.

A similar result was visible in (Gravanis et al., 2019). In this research, the classification accuracy of six machine learning algorithms was compared: K-NN, DT, Naïve Bayes (NB), SVM, AdaBoost and Bagging. All algorithms were tested using vector-based word representation extracted by Word2Vec (now referred to as Word2Vec). The highest classification accuracy achieved was 95% with the SVM using Word2Vec.

The SVM also achieved higher classification accuracy than the algorithms it was compared with in (Patwa et al., 2021). SVM using TF-IDF achieved 93.32% classification accuracy. TF-IDF was also used by three alternative algorithms. LR achieved 91.96% classification accuracy, DT 85.37% and Gradient Boost 86.96%.

(Verma et al., 2021) compared combinations of two different word embeddings and six machine learning algorithms. The word embeddings were TF-IDF and frequency-based word representations extracted by bag-of-words (now referred to as bag-of-words). The classification algorithms that were used for comparison were SVM, K-NN, NB, DT, Bagging and Adaboost. Using news content as input, SVM using bag-of-words achieved the highest classification accuracy with 95.61%.

The SVM and the LSVM show the most promising results for fake news detection. The word embeddings have however not been often compared. From the little comparison that has been done in the literature, bag-of-words shows the most promise. This has however not been compared to Word2Vec, which has achieved high classification accuracy for fake news detection research. This research adds to the literature by comparing these promising word embeddings and machine learning algorithms.

2.2.2 Deep Learning Classifiers

Neural networks are used for fake news detection in the article of (Wang, 2017). This article compares three models on a multilabel dataset, with two models containing news content as features. The first of these two models is a SVM using Word2Vec that achieves 25.5% classification accuracy. Next, a Convolutional Neural Network (CNN). This achieves a higher classification accuracy: 27%. Newer research from (Balwant, 2019) compares other neural networks on that same dataset. One of the tests is a Bidirectional Long Short-Term Memory (LSTM) network. It achieves a classification accuracy of 27.4%. This improved a base LSTM model in earlier research from (Long et al., 2017), which achieved 25.5%.

(Bahad et al., 2019) also compares neural networks, but on two different datasets. On both datasets, four different neural network architectures are tested: CNN, Recurrent Neural Network (RNN), Unidirectional LSTM-RNN and Bi-directional LSTM-RNN. On dataset 1, the Unidirectional LSTM-RNN achieves the highest classification accuracy with 91.48%. This is more than the Bi-directional LSTM-RNN that achieves 91.08% classification accuracy, the RNN which achieves 78.22% and the CNN which achieves 90.77%. On the second dataset, the scores look different. Here the Bi-directional LSTM-RNN achieves 98.75% classification accuracy, which is higher than the Unidirectional LSTM-RNN (98.63%), RNN (96.38%) and CNN (98.33%).

In (Kaliyar et al., 2020), four different machine learning algorithms using vector-based word embedding extracted by GloVe were tested against several neural networks. The algorithm with the highest classification accuracy was a Multinomial Naïve Bayes with 89.97%. What followed were a Decision Tree with 73.65%, a Random Forest with 71.34% and K-NN with 53.75%. However, the neural networks that were tested achieved higher classification accuracy. The CNN model achieved 91.50% and the RNN-LSTM model 97.25%. The one model that achieved even higher classification accuracy was the self-constructed alternative version of a CNN with 98.36%.

(Verma et al., 2022) tests a BERT-CNN hybrid model referred to as the MCred model on a dataset constructed out of four different datasets, of which the WELFake from Verma et al (2021) is one. It achieved a classification accuracy of 99.01%. This BERT-CNN hybrid achieves a higher classification accuracy than the BERT-RNN hybrid model (94.56%) and the BERT-LSTM model (96.94%).

These findings suggest that neural networks generally achieve higher classification accuracies when compared with machine learning algorithms. Neural networks have not been compared to machine learning algorithms on the WELFake dataset. This research aims to contribute to the scientific literature by incorporate both machine learning algorithms and neural networks in the experiments. Research from (Balwant, 2019) and (Bahad et al., 2019) showed high classification accuracies from the Bi-directional RNN-LSTM model. This research will test this model on the WELFake dataset and compare its classification accuracy to the machine learning algorithms.

2.3 Dataset

This section will address the datasets in fake news detection research. It will start off with a discussion on the datasets used in previous research projects. Then, it will move on to the dataset that will be used for this research.

(Shu et al., 2017) and (Horne & Adali, 2017) wanted to do research on detecting fake news using news propagation features. Therefore, they chose datasets that contained such features. Examples of these features are article engagements and social links. Because of the larger number of features in these datasets, they were unable to have as much news articles as other datasets that were constructed to detect fake news.

One of the datasets with a higher number of articles was the Liar Liar dataset. This dataset was labeled by PolitiFact and introduced in (Wang, 2017). It contains six labels, which is rarer than the more common binary labeling. The dataset contains 12,836 news stories, roughly balanced over the six labels. The features were part content related and part profile related. Although this dataset was substantially larger than its previous alternatives, other datasets with more news articles have now been published.

(Ahmed et al., 2017) introduces one of the larger datasets. This dataset contains news articles that were collected from the websites Reuters and Kaggle. This dataset contains 25,200 articles in total with text, type, title, date, and label as features. Since all true news came from the Reuters website and all fake news from Kaggle, the probability for bias in the fact checking is higher than in alternative datasets.

The UNBiased dataset in (Gravanis et al., 2019) aimed to specifically tackle the issue of bias by integrating several news sources and implementing other methods to avoid bias. It contains however only 3,004 news articles, making the dataset relatively small compared to the alternatives.

There are some other datasets that can be considered suitable for the task of content-based fake news detection. An example is the dataset from (Patwa et al., 2021) concerning COVID-19 tweets. However, it contains 10,700 tweets, making it smaller than alternatives.

Another dataset is the dataset used in (Kaliyar et al., 2020). This research used the Fake News dataset that is publicly available on Kaggle. It contains 20,800 news articles with their title, text, author, and label as feature. This dataset is used as one of several sub-datasets in the dataset of (Verma et al., 2021), which is the dataset this research will use.

(Verma et al., 2021) tested its experiments on the self-constructed WELFake dataset. The WELFake dataset contains 72,134 news articles originating from four different sub-datasets. The dataset contains news title, news content and label as features. There are multiple reasons why this dataset is the best choice for this research compared to its alternatives: it contains a relatively high number of news articles, it combines multiple datasets to reduce bias, it contains news content as feature, it is recent, publicly available and has been tested before. The WELFake dataset comes however with a limitation: the dataset uses binary labels, which might be an oversimplification of the issue. Multiple experiments have been performed on this dataset. However, promising results from models untested on the WELFake dataset leave room for possible improvement in classification accuracy. This research will perform these experiments on this dataset.

3 Methods

This section will discuss the composition of the methods that will be tested within this research. The discussed methods will be the feature extraction methods bag-of-words and Word2Vec and the Bi-directional RNN-LSTM architecture.

3.1 Methods for feature Extraction

The SVM is a binary classifier that attempts to fit the widest classification margin possible that separates the classes of data (Géron, 2019). Before the SVM classifies textual data, the words need to be represented using frequency-based or vector-based word representation. Bag-of-words is an example of frequency-based word representation. Bag-of-words operates by transforming the preprocessed words of an instance into a vector using one-hot-encoding. After this transformation, the vector can be fed to the model. Before the transformation of the instance can take place, a vocabulary must be built. Bag-of-words builds this vocabulary by fitting the preprocessed instances into the vocabulary (Manish, 2019).

In contrast with bag-of-words, Word2Vec feature extraction is a vector-based word representation. Word2Vec learns by minimizing the loss function. Word2Vec has the possibility of choosing between methods. Skip Gram and Continuous Bag Of Words (CBOW). Skip Gram predicts the context of a word using the word itself as input. CBOW takes the word context as input and predicts a word for that context (Minnaar, 2015).

3.2 Method for Bi-directional Recurrent Neural Network with Long Short-Term Memory

In this section the mathematical basis for the calculation of the recurrent layer and the LSTM architecture will be defined using the formulas from (Géron, 2019). The RNN is composed of recurrent neurons. Recurrent neurons are unique in the sense that they receive an input from the current instance and an output from previous instance(s). In this section, the workings of a recurrent neuron will be defined using this formula:

$$y_{(t)} = \phi(W_x{}^t\ X_{(t)} + W_y{}^t\ Y_{(t-1)} + b)$$

This formula demonstrates the calculations the output of a single recurrent neuron, $Y_{(t)}$. The t here stands for the current timestep. ϕ is the kernel function. What is defined here is thus the output for the current timestep. The phi symbol refers to the activation function that is utilized. The output layer in this experiment uses Sigmoid as an activation function, but the fourth (Dense) layer uses ReLU. Before applying the activation function, there are three variables summed up. The first is a multiplication of the weight vector of the input W_x at time step t with the input vector X at time step t. The second part is similar: a multiplication of the weight vector of the output W_y at time step t with the output vector Y at timestep t–1. The last part is adding the bias vector B. When these three are added together, the activation function is performed and the output Y at timestep t is produced. When this is done for all recurrent neurons, it leads to the following formula:

$$y = \phi(X_{(t)}W_{(x)} + Y_{(t-1)}W_y + b) = \phi([X_t\ Y_{(t-1)}]W + b)$$

$$With\ W = \begin{bmatrix} W_x \\ W_y \end{bmatrix}.$$

This formula uses the same definitions as the formula above. *Phi* means activation function, X the input vector, Y the output vector, W the weights, t the timestep, and B the bias vector.

The experiment will operationalize the Bi-directional RNN-LSTM architecture, which makes use of a Bi-directional LSTM layer. The Bi-directional LSTM layer is a regular LSTM layer where the input flows backward and forward, instead of one direction with the regular RNN-LSTM. The LSTM cell is split into two vectors: $h(t)$, which is the short-term state, and $c(t)$, which is the long-term state. The LSTM cell also has three gates: a forget gate, an input gate, and an output gate. The $c(t-1)$ vector drops memory by passing through a forget gate and adds memories that were selected by the input gate, resulting in the output of $c(t)$. Then, a copy of this passes through the output gate, which results into the new $h(t)$.

The $h(t-1)$ and $X(t)$ passes through four different fully connected layers, which produce four separate outputs: $f(t)$, $g(t)$, $i(t)$ and $o(t)$. The layer that outputs $g(t)$ uses the *tanh* activation function to produce what in the end becomes the output $y(t)$ and the new short-term state $h(t)$. The other fully connected layers use the logistic activation function to transform the inputs to an outputs between 0 and 1. These other fully connected layer outputs function as the three gates. This results in the following LSTM computations for the function that outputs $g(t)$:

$$g_{(t)} = tanh(W_{xg}\ ^{T}X_{(t)} + W_{hg}\ ^{t}h_{(t-1)} + b_g)$$

In this formula, the weight for the x vector passing to $g(t)$ called W_{xg} is multiplied with the input vector x at timestep t. Then it is added the weight vector for the short-term state passing to $g(t)$ at timestep t called $W_{hg}\ ^{t}$ that is multiplied with the short-term state h at timestep $t-1$. Lastly, it is added to the bias term of g called B_g.

The other gate functions operate using the same formula with some minor changes. Instead of the *tanh* activation function, the logistic activation function is used. The other difference is that wherever a g is in the formula for the $g_{(t)}$ output, the g changes to whatever output is produced. For example: for the $o_{(t)}$ output, the g variable is replaced with the o variable. For the $o_{(t)}$ output, this results in the following formula:

$$o_{(t)} = \sigma(W_{xo}\ ^{T}X_{(t)} + W_{ho}\ ^{t}h_{(t-1)} + b_o)$$

The $c_{(t)}$ output is calculated by an element-wise multiplication $f_{(t)}$ and $c_{(t-1)}$. Next, this is added to the element-wise multiplication of $i_{(t)}$ and $g_{(t)}$. This results in the following formula:

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)}$$

Finally, $y_{(t)}$ is the same as $h_{(t)}$ and thus calculated using the same formula: an element-wise multiplication of $o_{(t)}$ and a hyperbolic tangent activation function on $c_{(t)}$. This results on this formula:

$$y_{(t)} = h_{(t)} = o_{(t)} \otimes tanh(c_{(t)})$$

4 Experimental Setup

This section will go deeper into how the experiments were set up. It will do so by discussing the used dataset, the sampling and preprocessing of the data, the procedure of the experiments, the evaluation metrics that are used and finally the implementation details.

4.1 Dataset

The dataset used for the experiments in this research is the WELFake dataset from (Verma et al., 2021). The WELFake dataset is a merger of four popular news datasets. The sources of the four original datasets are Kaggle, McIntire, Reuters and Buzzfeed Political. The WELFake dataset contains 72.134 news articles with 35.028 articles classified as real and 37.106 articles classified as fake. The dataset has the article title and article text as features. 558 article titles are missing in the dataset, and 39 article texts. No news article has a missing true or false label. The dataset is freely available on Kaggle as a CSV file with a size of 245 MB.

4.2 Data Sampling

After downloading the dataset in a Pandas dataframe, the missing values in the data were dropped. Then the data was sampled using hold-out cross-validation. 80% of the data was assigned to be training data, and 20% was separated as test data. The choice for the hold-out cross-validation was based on it being faster than alternatives like k-fold cross validation. The speed was especially relevant for the more complex models, such as the Bi-directional RNN-LSTM. The splitting was done randomly because the dataset contains no features that could signal clusters in the data, such as a publishing date.

4.3 Data Preprocessing

After the data was split, the data was preprocessed. For natural language processing tasks, the goal of preprocessing the data is to allow the data to be vectorized so the model can process the data. Several steps were undertaken to reach this stage. The first step was removing punctuation from the news articles. Next, the text was tokenized. Tokenization is a process where the input text is split into smaller units. There are two types of tokenization: word tokenization and sentence tokenization (Hagiwara, 2021). This research uses word tokenization. After this, the tokens were lowered. Then commonly used stopwords in the tokens were dropped as they add very little value to the analysis and therefore carry little meaning (Deepanshi, 2022). Finally, the tokens went through a process of lemmatization. Lemmatization means transforming the word to its original form. For example: the lemma of “met” is “meet” (Hagiwara, 2021). The inspiration for the data preprocessing code came (Benicio, 2022) on Kaggle.

4.4 Experimental Procedure

The central task is binary classification of textual data. This section will address the modeling choices for the performed experiments. This will be done by discussing the hyperparameter tuning as well. An overview of the tuning results is included in appendix A. At the end, it will discuss the confusion matrix for the error analysis.

4.4.1 Set-up for the Support Vector Machine with frequency-based word representation extracted by bag-of-words

The best performing classifier on the WELFake dataset is a SVM using bag-of-words as feature extraction methods in (Verma et al., 2021). The original code has not been made available. Therefore, a new code has been produced for this experiment. The code for bag-of-words was based on a code from (Benicio, 2022) that also attempted to binary classify fake news articles.

The code for training a SVM model was gathered from (Géron, 2019). The kernel used was the default RBF kernel. The hyperparameters of the SVM were tuned using GridSearch from the SKLearn library: C and gamma. The grid search options for C were 0.001, 0.01, 0.1, 1 and 10. For Gamma, the grid search options were 1, 10, 50, 100 and "Scale". Because of the relatively large time needed for tuning, 2-fold cross-validation was used. After tuning, the hyperparameters that were used for training are 10 for C and "Scale" for Gamma.

4.4.2 Set-up for the Support Vector Machine with vector-based word representation extracted by Word2Vec

The first alternative model is a SVM using Word2Vec as feature extraction method. The choice for Word2Vec is a combination of the ease of implementation and the potential relevance of the method based on earlier research. The Word2Vec embedding was imported in python from the Gensim library. This research did not use a pretrained Word2Vec, but trained Word2Vec using the preprocessed data. For this training, the hyperparameter min count was set to 1 to not lose any data. The hyperparameters were imitated from the inspirational code from (Bijoy, 2020) and resulted in the following: a vector size of 300, skip-gram training algorithm, window of 5, workers of 4 and epochs 50.

The kernel options when tuning the SVM were "RBF", "Sigmoid" and "Linear." The hyperparameters C and Gamma of the SVM were also tuned by GridSearch. The options for C were 0.001, 0.01, 0.1, 1 and 10 and for Gamma Because training took less time, 5-fold cross-validation was used. The hyperparameters that were used for training are "RBF" as kernel, a C value of 10 and a Gamma of 0.1.

4.4.3 Set-up for the Linear Support Vector Machine with frequency-based word representation extracted by bag-of-words

Next, the LSVM using bag-of-words feature extraction. The bag-of-words implementation is the same as for the regular SVM. Two hyperparameters were tuned for the LSVM: the loss function and C. The candidates for the loss function were Hinge and Squared Hinge. For C there were six: 0.001, 0.01, 0.1, 1, 10 and 100. After 5-fold cross-validation, a C value of 0.01 and the Squared Hinge Loss function were used for training the LSVM.

4.4.4 Set-up for the Bi-directional Recurrent Neural Network with Long Short-Term Memory

The Bi-directional RNN-LSTM has 5 layers and one output layer. The first layer is an embedding layer which receives an input of a specified 512 dimensions and produces an output using 128 neurons. Then a Bi-directional LSTM layer follows with 128 neurons, a dropout value of 0.2 and a recurrent dropout value of 0.2. The third layer is a dropout layer with a rate of 0.5. Then the RNN continues with a fourth dense layer of 64 neurons and a ReLU activation function. The fifth layer is another dropout layer with a rate of 0.5. The RNN concludes with a dense output layer of one neuron. The model is compiled with the Adam optimizer, the binary crossentropy loss function and the accuracy metric. A full overview of the architecture is included in appendix B.

The architecture is based on two other codes that are publicly available that perform a similar task and achieved high classification accuracies. From (Jaikrishnan, 2019), the architecture of the neural network was used. This includes the number of layers, the layer types and the values within the different layers. The input dimensionality and the number of neurons on the layers was based on the code from (Mishinev, 2022).

The hyperparameters epochs and batch size were initially based on the code from (Mishinev, 2022) as well, with epochs at 8 and batch size at 64. Because of large time finetuning takes, only batch size was tuned using GridSearch from the candidates 64, 32 and 16 using 2-fold cross-validation. The final hyperparameter choice for batch size was 16. After a test run with the model, the validation accuracy kept going up after the 8 epochs. Therefore, it was increased to 20 epochs for the final model.

4.4.5 Confusion Matrix

For the second sub-question, a confusion matrix will be used for the error analysis for the model with the highest classification accuracy. This confusion matrix will be constructed for each of the four sub-datasets out of which the WELFake dataset is constructed. From these sub-datasets, the missing values will be dropped before applying a confusion matrix. Splitting the confusion matrix by the four sub-datasets will provide additional insights. The first sub-dataset is from Kaggle and contains 20664 news articles. Next is the McIntire sub-dataset, containing 6285 articles. Then the Reuters sub-dataset, with 44487 articles. The last sub-dataset is from Buzzfeed and contains 101 articles. The confusion matrix of the Scikit-Learn library will be used for this error analysis.

4.5 Evaluation Criteria

The main evaluation metric used to answer the research question is classification accuracy. Accuracy is the percentage of instances that the model predicted correctly. The benefit from this evaluation metric is that it is straightforward and easy to understand. The issue with accuracy is that it can be very good at predicting a dominant class, but bad at predicting a different class. This is especially common in unbalanced datasets.

To correct for this difficulty, three other evaluation metrics will be included: recall, precision and F1 score. Recall is true positives divided by the sum of true positives and false negatives. Precision is true positives divided by the sum of true positives and false positives. There is a trade-off between precision and recall. The F-measure balances these out as it finds the balance between precision and recall. These three metrics will be included on average and per label.

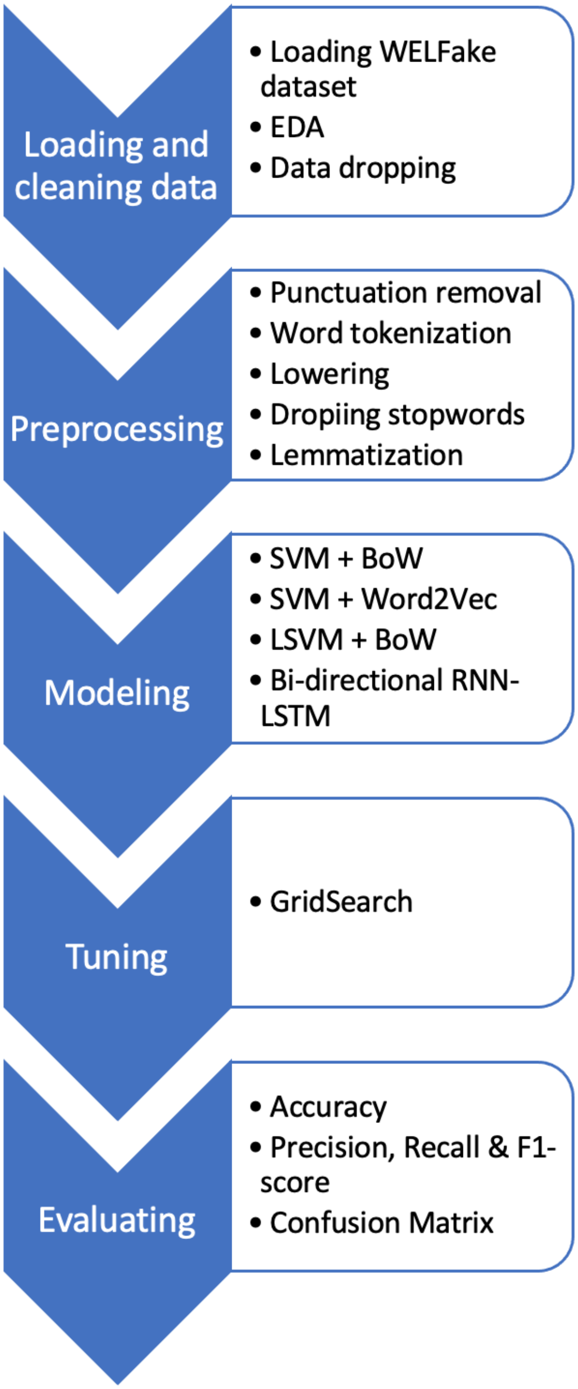
4.6 Implementation

All coding was done in Python. Every experiment was performed in Jupyter Notebook, which was accessed through Anaconda. The processor used is a Dual-Core Intel Core i5 with a processor speed of 1,6 GHz. It has 8 GB of RAM.

The coding used the following libraries: SKlearn, Gensim, Keras, Tensorflow, Pandas, Matplotlib and Seaborn.

An overview of the experiments is presented in the data science pipeline in figure 1

Figure 1: Pipeline.
Source: illustration by the author



5 Results

The results of the experiments will be discussed here. This will be split using the structure of the research questions. First, the classification accuracy of the different classifiers for fake news detection will be compared. Then, an error analysis of the confusion matrix will be processed. The information in the figures will be discussed in the text.

5.1 Classification Accuracies

The baseline model is the SVM with bag-of-words. The results of the experiments are collected in table 1. These results will be discussed in this section. All the percentages presented in the tables are the percentages that were obtained by testing on test data.

Table 1: Evaluation metrics for sub-question 1

	SVM + BoW	SVM + W2V	LSVM + BoW	Bi-directional RNN-LSTM
Accuracy	0.9698	0.9436	0.9677	0.9719
Precision	0.9702	0.9436	0.9679	0.9721
Recall	0.9696	0.9435	0.9675	0.9718
F1	0.9698	0.9436	0.9677	0.9719
Precision fake	0.9792	0.9450	0.9740	0.9774
Recall fake	0.9589	0.9405	0.9599	0.9649
F1 fake	0.9689	0.9424	0.9669	0.9711
Precision true	0.9611	0.9428	0.9618	0.9667
Recall true	0.9804	0.9466	0.9753	0.9786
F1 true	0.9706	0.9447	0.9685	0.9719

5.1.1 Results for the Support Vector Machine with frequency-based word representation extracted by bag-of-words

The SVM with bag-of-words feature extraction classifies with an accuracy of 96.98%. It achieves a precision of 97.02%, a recall of 96.96% and a F1 score of 96.98%. When purely classifying fake news, its precision is 97.92%, its recall 95.89% and its F1 score 96.89% When tasked with only true news, it has a precision of 96.11%, a recall of 98.04% and a F1 score of 97.06%

5.1.2 Results for the Support Vector Machine with vector-based word representation extracted by Word2Vec

The SVM with Word2Vec feature extraction has a classification accuracy of 94.36%. Its precision is 94.36%, recall 94.35% and F1 score 94.36%. When classifying fake news specifically, its precision rises to 94.50% but the recall and F1 score decrease to 94.05% and 94.24%. When classifying true news, precision is at 94.28%. The recall is then 94.66% and the F1 score 94.47%.

5.1.3 Results for the Linear Support Vector Machine with frequency-based word representation extracted by bag-of-words

The LSVM with bag-of-words feature extraction achieves a 96.77% classification accuracy. The other main metrics are not far apart, with 96.79% precision, 96.75% recall and 96.77% for its F1 score. For classifying fake news, its precision is 97.40%, its recall 95.99% and its F1 score 96.69%. For true news, it achieves 96.18% precision, 97.53% recall and a F1 score of 96.85%.

5.1.4 Results for the Bi-directional Recurrent Neural Network with Long Short-Term Memory

The Bi-directional RNN-LSTM classifies the news articles with 97.19% accuracy. The precision is 97.21%, its recall 97.18% and its F1 score 97.19%. For fake news specifically, its precision is 97.92%, its recall 95.89% and the F1 score 96.89%. When the neural network classifies true news, the precision is 96.11%, the recall 98.04% and the F1 score 97.06%. The accuracy and loss over epoch of the neural network for the training and test data is included in appendix C.

5.2 Error Analysis

The model that achieved the highest classification accuracy is the Bi-directional RNN-LSTM. In the second sub-question, a confusion matrix is used to analyze asymmetries or biases in the conclusion of this model split by sub-dataset. There are four sub-datasets. For each of these sub-datasets, the confusion matrix will be provided. The label 0 in the confusion matrix refers to fake news, the label 1 refers to true news.

5.2.1 Kaggle sub-dataset

Figure 2 shows the confusion matrix of the Kaggle sub-dataset. The Kaggle-sub dataset contains 20664 classified news articles. Of these news articles, 9990 news articles are labeled fake and 10674 news articles are labeled true. From these news articles, 9972 articles are predicted as fake and 10692 articles were predicted as true. There are 9899 true negatives, 73 fake negatives, 91 fake positives and 10601 true positives. This means 0.7947% of the Kaggle sub-dataset will not be predicted correctly by the best performing model.

Figure 2: Kaggle

	Predicted fake	Predicted true
Actual true	73	10601
Actual fake	9899	91

5.2.2 McIntire sub-dataset

Figure 3 shows the confusion matrix of the McIntire sub-dataset. This sub-dataset contains 6285 news articles in total. There are 3058 news articles labeled as fake and 3227 news articles as true. 3051 news articles are predicted as fake and 3234 articles are predicted as true. 3030 news article predictions are true negatives, 21 article predictions are fake negatives, 28 predictions are fake positives and 3206 articles are true positives. This means that 0.7796% of news articles are classified wrongly in the McIntire sub-dataset.

Figure 3: McIntire

	Predicted fake	Predicted true
Actual true	21	3206
Actual fake	3030	28

5.2.3 Reuters sub-dataset

Figure 4 is the confusion metrics for the Reuters sub-dataset. This sub-dataset is the largest with 44487 news articles. 21935 news articles are labeled as fake, while 22552 articles are labeled true. 21833 articles are predicted as fake while 22654 articles are predicted as true. 21709 predictions are true negatives, 124 predictions are fake negatives, 226 predictions are fake positives, and 22438 predictions are true positives. 0.7867% of the total predictions in the Reuters sub-dataset are incorrect.

Figure 4: Reuters

Actual true	124	22428
Actual fake	21709	226
	Predicted fake	Predicted true

5.2.4 Buzzfeed sub-dataset

Table 5 is the final confusion matrix, which is for the Buzzfeed sub-dataset. This is the smallest sub-dataset with only 101 news articles. There are 45 news articles that are labeled false and predicted as false and thus fully overlap. The same goes for the true news articles that are labeled as true and predicted as true. This means that there are 45 true negatives, 56 true positives and no fake negatives or fake positives. There are 0 news articles classified incorrectly in the Buzzfeed sub-dataset.

Figure 5: Buzzfeed

Actual true	0	56
Actual fake	45	0
	Predicted fake	Predicted true

6 Discussion

The goal of the research was to improve the classification accuracy over the state-of-the-art classifier from the scientific literature. This section will discuss the results of the experiments more in depth. It will start with a comparative discussion of the empirical results. Then it will continue to the degree of generalizability of this research.

6.1 Empirical Results

To answer the main research question, two sub-questions have been defined. These sub-questions will be answered in this section by discussing the results that have been collected from the performed experiments.

6.1.1 Discussion of the classification accuracies

The first sub question was: to what extent do a Support Vector Machine with vector-based word representation extracted by Word2Vec, a Linear Support Vector Machine with frequency-based word representation extracted by bag-of-words and a Bi-directional Recurrent Neural Network with Long Short-Term Memory improve classification accuracy over a Support Vector Machine with frequency-based word representation extracted by bag of words? To answer this, all classifiers were tested in python on the WELFake dataset. The results will be discussed in this section and compared to the results in related works.

The highest classification accuracy on the WELFake dataset was 95.61%. This accuracy came from a SVM with bag-of-words in (Verma et al., 2021). This research imitated this architecture as a baseline model but managed to get 96.98% accuracy. This difference is possible due to the differences in hyperparameters.

The 96.98% accuracy of the Support Vector Machine with frequency-based word representation extracted by bag of words was higher than the Support Vector Machine with vector-based word representation extracted by Word2Vec, which classified with 94.36% accuracy. The Support Vector Machine with vector-based word representation extracted by Word2Vec has been tested before in fake news detection in the research of (Gravanis et al., 2019). On the UNBiased dataset, it achieved a classification accuracy of 95%. However, the article of (Gravanis et al., 2019) did not compare bag-of-words feature extraction and Word2Vec feature extraction. The results of the experiments in this research indicate that frequency-based word representation has more potential for effectively classifying fake news articles than vector-based word representation when classifying with a Support Vector Machine.

The Support Vector Machine was compared with a Linear Support Vector Machine on the task of fake news detection in (Ahmed et al., 2017). Here, the Linear Support Vector Machine achieved a classification accuracy of 92%. This was higher than the 86% of the Support Vector Machine. The authors concluded that linear-based classifiers achieved better results than nonlinear ones. Both used TF-IDF for feature extraction instead of bag-of-words. This research experiments with a Support Vector Machine and a Linear Support Vector Machine with frequency-based word representation extracted by bag-of-words. When comparing the results, the Support Vector Machine classifies better than the Linear Support Vector Machine with 96.98% accuracy compared to 96.77%. This contrasts the claim made by the authors of (Ahmed et al., 2017) that linear-based classifiers achieve better results than nonlinear classifiers in fake news detection.

The best performing classifier that has been tested in this research is the Bi-directional Recurrent Neural Network with long short-term memory. This neural network classifies with an accuracy of 97.19% and achieves therefore higher accuracy than its Support Vector Machine alternatives. This result is similar to the results in (Balwant, 2019), where the Bi-directional Recurrent Neural Network with long short-term memory was also compared to a Support Vector Machine. Here, the Bi-directional Recurrent Neural Network with long short-term memory achieved a classification accuracy of 27.4% on the Liar-Liar dataset with six classes. This was compared to the 25.8% accuracy of a Support Vector Machine on the Liar-Liar dataset in (Wang, 2017). The Bi-directional Recurrent Neural Network with long short-term memory has also been tested on a binary labeled dataset in the research of (Bahad et al., 2019), where it achieved 91.08% and 98.75% accuracy. In this research it was however not compared to machine learning classifiers. This comparison is the contribution of this research. The results support the general pattern in the related works on fake news detection, where neural networks achieve a higher classification accuracy than machine learning classifiers.

6.1.2 Discussion of the error analysis

The second research question is: to what extent are there asymmetries/biases in the conclusion from the confusion matrix in the model that performs with the highest classification accuracy split by sub-dataset? This will be answered for the Bi-directional Recurrent Neural Network with long short-term memory, which achieved the highest classification accuracies with 97.19%. This analysis is done with the confusion matrix of each of the four sub-datasets out of which the WELFake dataset was constructed.

One of the confusion matrices seems of little value for insights: the Buzzfeed sub-dataset, which is the fourth and last one. The sample size from the Buzzfeed dataset is relatively small with only 101 news articles. In the Buzzfeed confusion matrix, there are zero incorrect classifications. However, because of the small sample size there cannot be any conclusions drawn from this result.

This is not the case for the other three sub-datasets. Although these datasets differ in size as well, they all contain substantially more news articles than the Buzzfeed sub-dataset. To illustrate: the second smallest sub-dataset, the McIntire sub-dataset, contains 6285 articles.

Analyzing the confusion matrices of the three sub-datasets, the number of false positives is disproportionately large compared to the number of false negatives. All three sub-datasets have more false positives than false negatives. The proportion of false positives to false negatives is highest for the largest sub-dataset, the Reuters sub-dataset, where the number of false positives is almost twice as large as the number of false negatives.

The proportion of news articles in the whole WELFake dataset is a little skewed towards true news. However, the proportion of false positives to false negatives is larger than that for every sub-dataset except the small Buzzfeed sub-dataset. This leads to the conclusion that the model is biased toward predicting news articles as true.

6.2 Generalization

The introduction explained that fake news detection aims to tackle the spread of fake news. The Bi-directional RNN-LSTM classifier could be implemented on social media to alert individuals of possible falsities in the information. However, the detection strategy could be dangerous as well. (Tandoc Jr, 2019) stated corrections could backfire and increase believe in the wrong information. To make sure that fake news detection tackles the spread of fake news constructively, and therefore is of societal benefit, a focus on optimizing trust in fake news detection methods might be crucial. To achieve this trust, misclassifications must be limited. There are two potential directions for further limiting errors: increasing the classification accuracy and improving the dataset. This section briefly addresses these and points to potential further research on these topics.

First, the classification accuracy. The Bi-directional RNN-LSTM achieved a classification accuracy of 97.19%. This can be further improved by hyperparameter tuning, which has only been performed on the batch size in this research because of time constraints. Next, the SVM using Word2Vec combination did not use a pretrained Word2Vec. Experiments can be performed to potentially increase classification accuracy with a pretrained version of Word2Vec. Finally, the BERT-CNN hybrid from (Verma et al., 2022) achieved a high classification accuracy on a larger dataset but has not been tested on this dataset due to its complicated nature and its time and demand. This could potentially do well in this task.

For improving the dataset, three directions could be promising. The first direction is the features in the dataset. The article of (Verma et al., 2022) identified two other types of features for fake news detection in addition to its content: profile-based features and propagation-based features. When these would be added to a large dataset, it could potentially help detect fake news more accurately. Next, the topics of the news. In the WELFake dataset, a lot of the topics concern politics. To prevent errors being made when the model is generalized to other topics, it would be useful to have a larger variation of topics to train the model on. The last large improvement is to make the labeling multiclass. Since binary categorizing news content as either true or false can be considered an oversimplification of the issue, training a model on a dataset containing multiclass labeling might help prevent undesirable situations.

7 Conclusion

Fake news misinforms news consumers and can potentially cause dangerous situations. The article of (Bergstrom & West, 2021) named technological advancement as one of the strategies to tackle the spread of fake news. Data scientists address this by performing experiments on the task of fake news detection. This research adds to the literature by comparing promising models from previous research projects on fake news detection.

The research question of this research was stated in the introduction: how accurately can news articles be classified as fake news by their content using machine learning and deep learning techniques? This is answered using two sub-questions.

The first sub-question was: to what extent do a Support Vector Machine with vector-based word representation extracted by Word2Vec, a Linear Support Vector Machine with frequency-based word representation extracted by bag-of-words and a Bi-directional Recurrent Neural Network with Long Short-Term Memory improve classification accuracy over a Support Vector Machine with frequency-based word representation extracted by bag of words? This research compares the classification accuracies on the WELFake dataset from (Verma et al., 2021). The highest classification accuracy was 97.19%, achieved by a Bi-directional RNN-LSTM.

The classification accuracy of the Bi-directional RNN-LSTM is also higheer than than the previous highest classification accuracy achieved on the WELFake dataset. The previous highest classification accuracy was the Support Vector Machine using frequency-based word representation extracted by bag-of-words as constructed in (Verma et al., 2021), which classified with an accuracy of 95.61%.

The second sub-question was: to what extent are there asymmetries/biases in the conclusion from the confusion matrix in the model that performs with the highest classification accuracy split by sub-dataset? The error-analysis shows that the Bi-directional RNN-LSTM is biased towards predicting news as true.

Before the Bi-directional RNN-LSTM model would be implemented on a larger scale, it is necessary to think about the impact it would have. Fake news detection models can only have a positive impact on society when news consumers decide to trust the information coming from these models. It needs to be sure that this trust will not be unnecessarily damaged by misclassifications. To prevent these misclassifications, further research is required. A promising direction is improving the dataset for training the model, for example by multiclass labeling instead of binary labeling. Next, research needs to continue increasing the classification accuracy of promising models.

References

Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In I. Traore, I. Woungang, & A. Awad (Eds.), *Intelligent, secure, and dependable systems in distributed and cloud environments* (pp. 127–138). Cham: Springer International Publishing.

Bahad, P., Saxena, P., & Kamal, R. (2019). Fake news detection using bidirectional lstm-recurrent neural network. *Procedia Computer Science*, *165*, 74-82. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050920300806 (2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019) doi: https://doi.org/10.1016/j.procs.2020.01.072

Balwant, M. K. (2019). Bidirectional lstm based on pos tags and cnn architecture for fake news detection. In *2019 10th international conference on computing, communication and networking technologies (icccnt)* (p. 1-6). doi: 10.1109/ICCCNT45670.2019.8944460

Benicio, B. (2022). Fake news prediction.. https://www.kaggle.com/code/ciobeni/fake-news-prediction.

Bergstrom, C. T., & West, J. D. (2021). *Calling bullshit: the art of skepticism in a data-driven world.* Random House Trade Paperbacks.

Bijoy, B. S. (2020). *Binary text classification word2vec svm.* https://github.com/BIJOY-SUST/Binary-Text-Classification--Word2vec-SVM. GitHub.

Brashier, N. M., Pennycook, G., Berinsky, A. J., & Rand, D. G. (2021). Timing matters when correcting fake news. *Proceedings of the National Academy of Sciences*, *118*(5), e2020043118.

Bryanov, K., & Vziatysheva, V. (2021). Determinants of individuals' belief in fake news: A scoping review determinants of belief in fake news. *PLoS one*, *16*(6), e0253717.

Deepanshi. (2022). Text preprocessing in nlp with python codes.. https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/.

Di Domenico, G., Sit, J., Ishizaka, A., & Nunan, D. (2021). Fake news, social media and marketing: A systematic review. *Journal of Business Research*, *124*, 329–341.

Egelhofer, J. L., & Lecheler, S. (2019). Fake news as a two-dimensional phenomenon: A framework and research agenda. *Annals of the International Communication Association*, *43*(2), 97–116.

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Unsupervised learning techniques.* O'Reilly Media, Incorporated.

Giachanou, A., & Rosso, P. (2020). The battle against online harmful information: The cases of fake news and hate speech. In *Proceedings of the 29th acm international conference on information & knowledge management* (pp. 3503–3504).

Gravanis, G., Vakali, A., Diamantaras, K., & Karadais, P. (2019). Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications*, *128*, 201–213.

Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., & Lazer, D. (2019). Fake news on twitter during the 2016 us presidential election. *Science*, *363*(6425), 374–378.

Hagiwara, M. (2021). *Real-world natural language processing: Practical applications with deep learning.* Simon and Schuster.

Horne, B. D., & Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Eleventh international aaai conference on web and social media.*

Jaikrishnan, S. V. J. (2019). Fakenews detection using lstm neural network.. https://https://www.kaggle.com/code/jsvishnuj/fakenews-detection-using-lstm-neural-network/notebook.

Kaliyar, R. K., Goswami, A., Narang, P., & Sinha, S. (2020). Fndnet–a deep convolutional neural network for fake news detection. *Cognitive Systems Research*, *61*, 32–44.

Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., … others (2018). The science of fake news. *Science*, *359*(6380), 1094–1096.

Long, Y., Lu, Q., Xiang, R., Li, M., & Huang, C.-R. (2017). Fake news detection through multi-perspective speaker profiles. In *Proceedings of the eighth international joint conference on natural language processing (volume 2: Short papers)* (pp. 252–256).

Manish. (2019). Bag of words – count vectorizer.. https://excellencetechnologies.in/blog/bag-of-words-count-vectorizer/.

Minnaar, A. (2015). Word2vec tutorial part ii: The continuous bag-of-words model.. http://alexminnaar.com/2015/05/18/word2vec-tutorial-continuousbow.html/.

Mishinev, T. (2022). Fake news lstm baseline 97% accuracy.. https://www.kaggle.com/code/tmishinev/fake-news-lstm-baseline-97-accuracy/notebook/.

Moravec, P., Minas, R., & Dennis, A. R. (2018). Fake news on social media: People believe what they want to believe when it makes no sense at all..

Mourão, R. R., & Robertson, C. T. (2019). Fake news as discursive integration: An analysis of sites that publish false, misleading, hyperpartisan and sensational information. *Journalism studies*, *20*(14), 2077–2095.

Patwa, P., Sharma, S., Pykl, S., Guptha, V., Kumari, G., Akhtar, M. S., ... Chakraborty, T. (2021). Fighting an infodemic: Covid-19 fake news dataset. In *International workshop on combating on line ho st ile posts in regional languages dur ing emerge ncy si tuation* (pp. 21–29).

Pennycook, G., McPhetres, J., Zhang, Y., Lu, J. G., & Rand, D. G. (2020). Fighting covid-19 misinformation on social media: Experimental evidence for a scalable accuracy-nudge intervention. *Psychological science*, *31*(7), 770–780.

Pennycook, G., & Rand, D. G. (2020). Who falls for fake news? the roles of bullshit receptivity, overclaiming, familiarity, and analytic thinking. *Journal of personality*, *88*(2), 185–200.

Pennycook, G., & Rand, D. G. (2021). The psychology of fake news. *Trends in cognitive sciences*, *25*(5), 388–402.

Shu, K., Wang, S., & Liu, H. (2017). Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, *8*.

Tandoc Jr, E. C. (2019). The facts of fake news: A research review. *Sociology Compass*, *13*(9), e12724.

Verma, P. K., Agrawal, P., Amorim, I., & Prodan, R. (2021). Welfake: word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems*, *8*(4), 881–893.

Verma, P. K., Agrawal, P., Madaan, V., & Prodan, R. (2022). Mcred: multimodal message credibility for fake news detection using bert and cnn. *Journal of Ambient Intelligence and Humanized Computing*, 1–13.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *science*, *359*(6380), 1146–1151.

Walker, M., & Matsa, K. E. (2021). *News consumption across social media in 2021.* Retrieved from https://www.pewresearch.org/journalism/2021/09/20/news-consumption-across-social-media-in-2021

Wang, W. Y. (2017). " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Appendix

Appendix A

Tested hyperparameters with adopted values

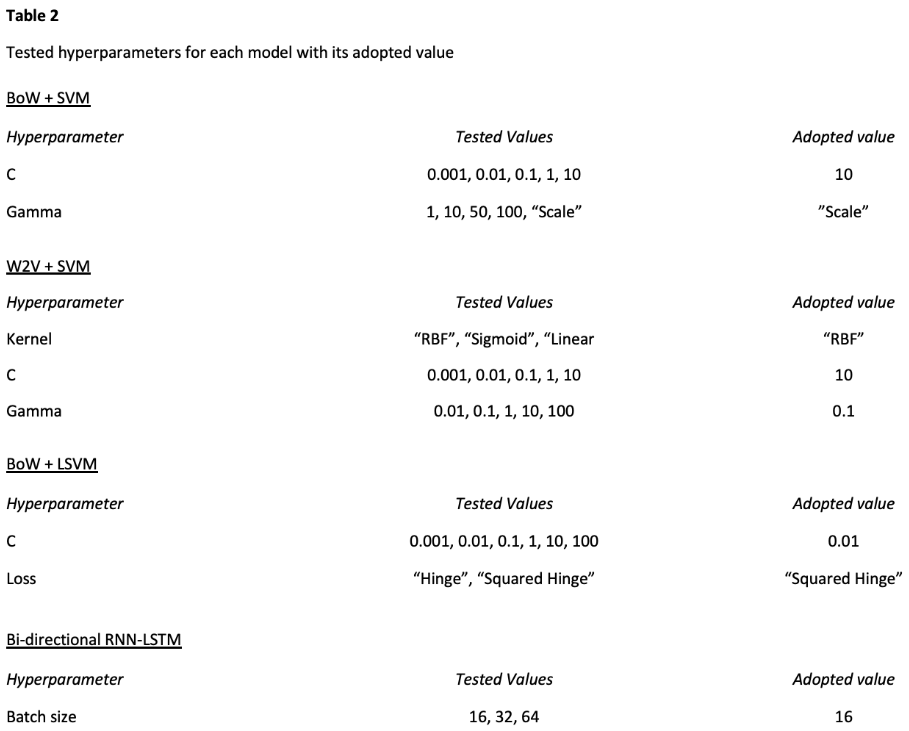


Table 2

Tested hyperparameters for each model with its adopted value

BoW + SVM

Hyperparameter	*Tested Values*	*Adopted value*
C	0.001, 0.01, 0.1, 1, 10	10
Gamma	1, 10, 50, 100, “Scale”	”Scale”

W2V + SVM

Hyperparameter	*Tested Values*	*Adopted value*
Kernel	“RBF”, “Sigmoid”, “Linear	“RBF”
C	0.001, 0.01, 0.1, 1, 10	10
Gamma	0.01, 0.1, 1, 10, 100	0.1

BoW + LSVM

Hyperparameter	*Tested Values*	*Adopted value*
C	0.001, 0.01, 0.1, 1, 10, 100	0.01
Loss	“Hinge”, “Squared Hinge”	“Squared Hinge”

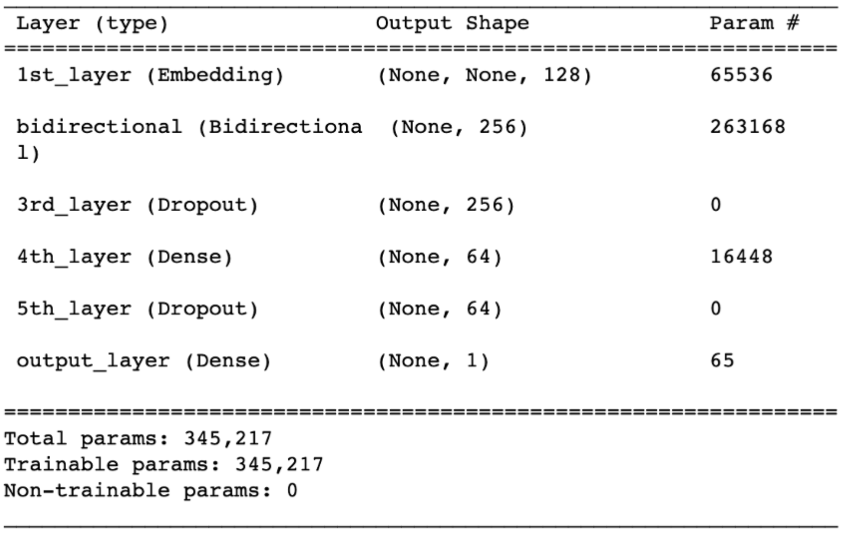
Bi-directional RNN-LSTM

Hyperparameter	*Tested Values*	*Adopted value*
Batch size	16, 32, 64	16

Appendix B

Neural Network Architecture

Figure 6
Architecture of the Bi-directional RNN-LSTM



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 1st_layer (Embedding)       (None, None, 128)         65536

 bidirectional (Bidirectiona  (None, 256)              263168
 l)

 3rd_layer (Dropout)         (None, 256)               0

 4th_layer (Dense)           (None, 64)                16448

 5th_layer (Dropout)         (None, 64)                0

 output_layer (Dense)        (None, 1)                 65

=================================================================
Total params: 345,217
Trainable params: 345,217
Non-trainable params: 0
_________________________________________________________________
```

Appendix C

Graphics of the Neural Network per epoch

Figure 7
Loss of the Bi-directional RNN-LSTM per epoch

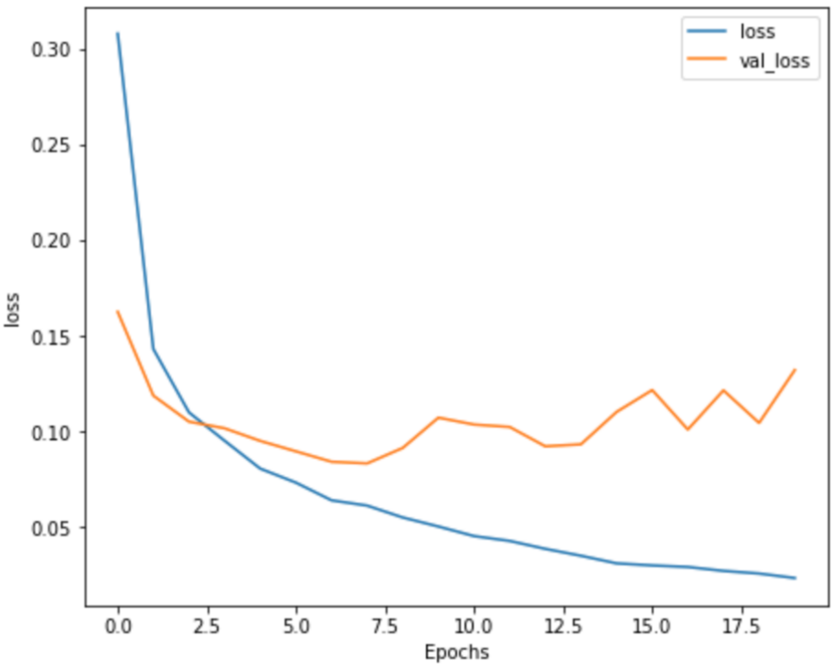
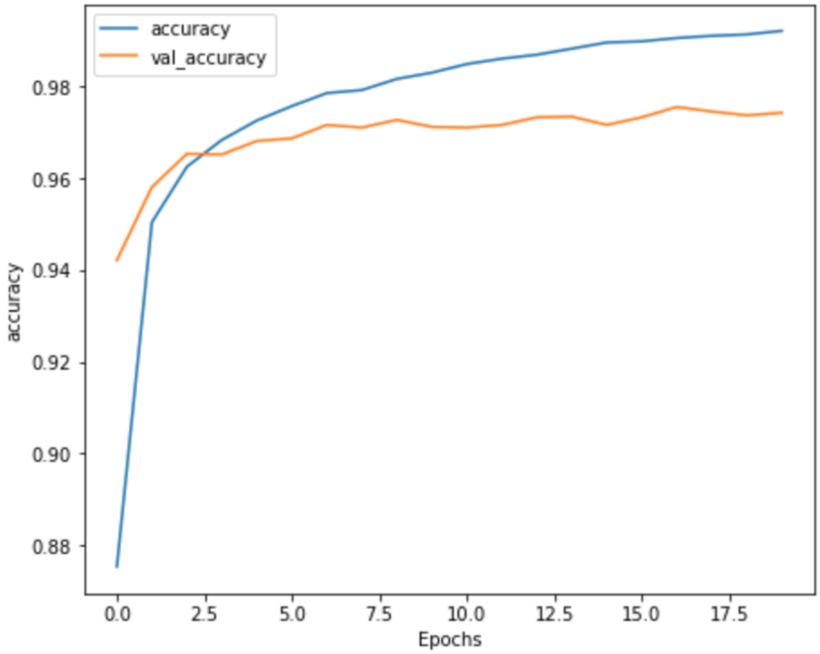


Figure 8
Accuracy of the Bi-directional RNN-LSTM per epoch


accuracy
val_accuracy
accuracy
Epochs