Otkrivanje lažnih vijesti u diskursu o ekologiji i klimatskim promjenama

Hanić, Sanja

Master's thesis / Diplomski rad

2024

UNIVERSITY OF ZAGREB

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 468

DETECTING FAKE NEWS IN ECOLOGY AND CLIMATE CHANGE DISCOURSE

Sanja Hanić

Zagreb, June 2024

UNIVERSITY OF ZAGREB

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 468

DETECTING FAKE NEWS IN ECOLOGY AND CLIMATE CHANGE DISCOURSE

Sanja Hanić

Zagreb, June 2024

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 04 March 2024

MASTER THESIS ASSIGNMENT No. 468

Student:	**Sanja Hanić (0036502476)**
Study:	Computing
Profile:	Computer Science
Mentor:	assoc. prof. Marina Bagić Babac

Title: **Detecting Fake News in Ecology and Climate Change Discourse**

Description:

The spread of fake news poses a significant challenge, particularly in the domain of ecology and climate change. This thesis employs Natural Language Processing (NLP) models and machine learning techniques to detect fake news articles within the ecological and climate change discourse. Web data will be scraped and collected into a novel dataset, augmented by existing datasets related to this topic, alongside official climate data. Through the application of advanced NLP algorithms, including sentiment analysis, topic modeling, and linguistic pattern recognition, the research aims to distinguish between authentic and misleading information. This research seeks to enhance the accuracy and reliability of fake news detection, contributing to the integrity of ecological and climate change communication.

Submission date: 28 June 2024

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 468

Pristupnica:	**Sanja Hanić (0036502476)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentorica:	izv. prof. dr. sc. Marina Bagić Babac

Zadatak: **Otkrivanje lažnih vijesti u diskursu o ekologiji i klimatskim promjenama**

Opis zadatka:

Širenje lažnih vijesti predstavlja značajan izazov, posebice u domeni ekologije i klimatskih promjena. Ovaj diplomski rad koristit će modele obrade prirodnog jezika (NLP) i tehnike strojnog učenja za otkrivanje lažnih vijesti unutar diskursa o ekološkim i klimatskim promjenama. Podaci s weba bit će izdvojeni i prikupljeni u novi skup podataka, proširen postojećim skupovima podataka koji se odnose na ovu temu, uz službene klimatske podatke. Primjenom naprednih NLP algoritama, uključujući analizu osjećaja, modeliranje tema i prepoznavanje jezičnih obrazaca, istraživanje ima za cilj razlikovati autentične informacije od pogrešnih.

Rok za predaju rada: 28. lipnja 2024.

Contents

Introduction

One of the catalysts of the rapid fake news sharing could be online communication channels. Author Nagi [1] says online tools offer accessible and easy sharing that drives the spread of fake news. Disinformation, misinformation and fake news are terms that are often used interchangeably, but many researchers point out the important differences between the three. Disinformation can be defined as „all forms of false, inaccurate or misleading information designed, presented and promoted to intentionally cause harm or for profit" [33]. Whereas misinformation is misleading and incorrect information that is shared without any harmful motives. Fake news articles and posts can be a combination of both disinformation and misinformation.

Even though the reports from Intergovermtal Panel on Climate Change (IPCC) urges that climate change is very real, climate change was found to be a polarizing issue for twitter users by researches [43]. Climate change is a polarizing issue on twitter where researchers have found that most users can be classified as believers or deniers [44]. It's not just the users on twitter that are not convinced climate change is a pressing issue, researchers point out that some individuals strongly believe it is a hoax [29]. Another issue that arises from climate change denialism is that „deniers" would have a more polarized opinion on public policies as solutions to climate change [34]. Clearly understanding the internet users perception on climate change seems to be an impossible task, since posts and news are created daily and can't be analiysed as quickly as they are produced. Looking at different studies we can gain perspective about users of certain platforms in the times of creating a dataset. A study on Twitter users opinions on climate change in 2016 concluded that there were more positive tweets than negative ones, but the neutral tweets made up the biggest group of tweets in the dataset [36]. On the other hand, another study about climate change tweets reported that 55.8% of the top 500 tweets by the number of reports in their dataset were denying climate change all together or denying that humans caused it [38]. One reason of such wide climate change denial presented by multiple researchers is the use of denial think tanks or denial machines. Researchers [32] stress the fact that in domain of tobacco industry and climate change multiple firms have deliberately manufactured doubt to downplay the actual effect they can have on our health and environment. Students opinions and worries about climate change were presented in another study and found that most students questioned (over 60%) were aware that climate change is real and that they do care about it personally [37]. A worrying effect of a monological belief system was found by researchers [31] in the domain of Covid-19, artificial intelligence and climate change where people who believe one fake news article are susceptible to believing other fake news as well. One positive aspect is the ability, or lack thereof, of fake news articles about climate to significantly influence climate skepticism. Researchers [35] found that the exposure to fake news regarding climate change had limiting effect on the persons trust in science.

This thesis explores how machine learning models can be used to automaticly detect fake news in the domain of climate change, witch is a global issue that is polarizing people. Multiple models are tested on a dataset containing real and fake news articles with additional features. The chapters of this thesis are as follows: the literature review chapter presents the current state-of-the art models and approaches to automatic fake news detection in different domains, the models chapter explains the models that were trained and tested for this thesis, the method chapter containes information about the used technologies such as relevant libraries and software and explains in detail how the text pre-processing, feature selection, word representation and model training was performed, the results chapter clearly presents the obtained results in multiple tables and figures, the discussion chapter comments and compares the results with similar studies and methods and points out the drawbacks and limitation of this thesis as well as possibilites for future improvements.

1. Literature review

Looking at recent systematic mapping studies (SMS) we can gain perspective about the current research of a topic and its development over years. Researchers conducted an SMS study where they mapped and counted relevant research regarding fake news detection from January 2010 until July 2021. In their work researchers [2] looked at 76 studies and concluded that future work should only focus on research from 2018 until today, since more than 50% of papers they included were published after 2018. Another advice from their SMS study is to include implementation of the proposed solutions for fake news detection, which is important for replicability of results. Their research also pointed out that the domain of fake news detection should be more diverse, since most papers cover politics, health and e-commerce. This is supported and advised by other researchers as well [3], therefore our work aims to fill this gap by providing a look into climate change and ecology specific fake news detection which can also fall into the scientific news category. Deep learning was found to be the most commonly used method for fake news detection by the same researchers although they criticize its „black box“ problem and lack of interpretability of these methods.

In their systematic literature review (SLR) researchers [3] gathered a total of 49 relevant fake news detection papers from 2017 until mid-august of 2021. They presented the most used machine learning methods and approaches by using a formal methodology for computer science research. They found support vector machine (SVM) to be the most common machine learning method used, although the best performing algorithms were random forests with an accuracy of 99.3%, followed by decision trees and Bayes theorem. Regarding deep learning, the best performing models found was a generic neural network without specification that achieved accuracy of 99.9% followed by convolutional neural networks (CNN). Researches also highlight that using hybrid deep learning models can improve accuracy significantly, for example using a pretrained bidirectional encoding representation transformer (BERT) as an embedding layer in a neural network. However, the same researchers emphasize that the accuracy achieved relies directly on the dataset used in training. Choosing the right features to describe the dataset can be vital and the most commonly used ones are term frequency (TF), term frequency inverse document frequency (TF-IDF), global vectors (GloVe), Bag-Of-Words, N-grams and CountVectorizer embeddings. This SLR also looked at what software is used to build fake news detection models and it was found that Python is the most popular tool for developing artificial inteligence (AI) solutions. Python has many libraries for machine learning and the most commonly used ones are sci-kit learn, Keras and TensorFlow. Researches also found that the most frequent domain of the fake news detection tasks are politics.

In their detailed review of fake news detection with deep learning, researchers [4] found four primary impacts that fake news have on society. They list the most important ways fake news can influence our lives: the impact fake news can have on individuals who are bullied online, the impact on people's well-being due to searching for health advice online, the impact on customers and businesses due to fake reviews and finally a democratic impact on voters as evidenced by the infamous 2016 US presidential election, which is speculated to have been highly influenced by misinformation [5].

In their survey on natural language processing for fake news detection researchers [6] emphasize the need for multiple levels of truth instead of using binary classification. Another area of interest is classifying entire articles instead of short claims. Researchers point out that this is a challenging task since expert annotation would be lengthy and assuming all articles from one news source are real or fake can be unlikely and dangerous. Researchers recommend investigating using hand crafted features in combinations with neural networks and the proper usage of non textual news data such as images and video.

Researchers compared different methods of machine learning: random forests, support vector machines, naive Bayes, logistic regression and gradient boosting. They found that the best result was achieved when using random forests with an accuracy of 98.3% and this result was improved by using ensemble learning which enables them to use multiple models at once [7]. Ensemble learning allows each machine learning model included to vote on a given task. Their results were comparable with deep learning models, even when using simpler machine learning models which could be due to ensemble learning.

Researchers explored "reality vertigo" as a problem that fake news causes within our society [8]. They compared multiple machine learning models and multiple vectorization models and they recommended using term frequency - inverse document frequency TF-IDF vectorization for its simplicity and high accuracy. The best performing model they found was a convolutional neural network but long training time is one of its minuses.

Researchers present multiple directions for future work and highlight the need for standard dataset and evaluation metrics.

One researcher presented the differences in style of fake news compared to real news, their methods were only based on stylometric features [9]. A stylometric classifier that was used is a logistic regression model that has features based on n-grams, part of speech (POS) tags and dictionaries. Even though the proposed stylometric classifier managed to learn and improve it did not have an advantage compared to the state-of-the-art models. Author points out that stylometric differences might be too subtle in contrast to other more defining features, but it is a promising approach for the fake news detection task.

A novel approach was described by researchers [10] that modeled a tri-relationship between news publishers, news content and users. To create a news latent stance multiple features were used in their framework TriFN : a news feature embedding, a user embedding that models user-user interaction, user credibility, user-news engagement and user latent stance, another embedding was a publisher-news link and publisher partisan embedding. These components were obtained through semi-supervised feature learning. The framework had a F1 score of more than 80% within 48 hours on the datasets used and this result is promising for resolving fake news in the earliest stages. Embedding social context into this framework could have improved the results especially for early stage fake news detection.

Another research that took into consideration the user interaction looked at both components of news content and comments written by users in regards to that article. Researchers [11] showed that user comments improve detection performance significantly. The framework used is called dEFEND and it is a deep hierarchical co-attention network.

Authors contributed to the fake news detection work by presenting their hybrid CNN-RNN model in detail to ensure reproducibility [12]. They showed that a hybrid CNN-RNN approach tends to work well on a specific dataset however its performance is less than ideal on unseen data, hence it does not generalize well.

Authors [13] collected their own dataset in German by labeling unreliable sources as fake and having a similar approach for true articles. Their COVID-19 fake news dataset also included tweets that were connected to news articles. Their methods included the term frequency minus inverse document frequency (TF-IDF), CoCoGen, BERT and a BERT + social context model which was found to be the best performing model in their work.

Authors emphasize that having a specific source of news can cause the model to learn stylistic features of that source instead of learning how to solve a classification task. The problem of models learning specific styles of news sources due to a lack of variation of sources was also noticed by the authot in their work [9]. Authors criticize the use of "black box" models since they are not interpretable and lack the ability to solve practical problems in industries that need to explain results such as forensic and medicine [13]. Their results show that using simpler language features such as TF-IDF and social context can also prove to be valuable in the fake news detection task.

From bag-of-words approaches to deep learning there is a wide range of fake news detection methods. In this chapter I will present the current state of the literature regarding fake news detection to the best of my abilities. Automatic or computational fake news detection has been a topic of growing interest for researchers in the last years, especially after the 2016 US Presidential elections where the problem of fake news has been recognized as a public concern [5]. The growing mass of research emphasizes the need for automatic fake news detection since an average reader can't always distinguish between real and fake news.

An interesting study by researchers [30] found that a third-person perspective was identified in the dissemination of fake news about global warming, meaning that most people felt more confident in their own ability to recognise fake news about global warming than others. Weather people are cofident in their ability to recognise fake news or not, some studies have shown that people struggle with recognising fake news. Multiple studies have tested computational methods vs. humans in fake news detection and found the former to be more successful [14], [15], [16], [17], [18]. In their study on human and algorithmic detection of fake news, researchers point out that humans recognized fake news correctly 64% of the time, while the algorithms used outperformed human readers by achieving a result of 67% accuracy [14]. Somewhat more impressive results were obtained by authors in [15], where their RF model predicted fake news in Arabic with a 87% accuracy, while humans achieved a 78% accuracy. When looking at fake online reviews researched [16] have concluded that humans can accurately recognise fake reviews 57% of the time, while their automated approach achieved an accuracy of 81%, it is important to mention that this research looked at fake reviews and not fake news since there might be differences in human perception of online reviews and news. As with the previously mentioned research the human annotators in the study done by [17] were not experts and they underperformed with accuracy of 88% when compared to their best performing model SVM with accuracy of 94% in the task of fake, satirical and real news detection. Even when we look at research from [18] we can see that humans recognized fake news 50%-63% of the time, depending on the setting, and their machine learning algorithm recognized fake news accurately 65% of the time. On the other hand, several studies have shown humans outperform machines in certain fake news detection tasks [19], [20]. In their work on machine generated fake news detection humans have outperformed their stylometry-based classifier, and human results were significantly improved when allowed to use external sources to verify or disprove of the fake news articles [19]. When comparing fake news content in the celebrity domain and more serious and varying news domain researchers found that humans outperformed their system only in the former domain [20].

As with many NLP tasks BERT based models emerge as the best options when it comes to fake news detection [21]. Researchers that used BERT based models showed that they outperform other methods [22], [23], [24], [25], [26]. In their survey of studies that used BERT based models, researchers [21] concluded that BERT has become a baseline for all NLP tasks. Researchers [22] proposed a model called FakeBERT that uses word-embeddings from BERT and a complex neural network with multiple layers to achieve an accuracy of 98.9 % on the task of fake news detection. In their study on French fake news detection researchers [23] obtained an F1 score of 84.75 by using a BERT model with 6 hand crafted linguistic features and therefore outperformed other ML approaches such as SVM, MNB and bag-of-words approaches. The hand crafted features included length, ratio of adverbs and numbers, ratio of terms that express modality, Flesch-Kincaid Reading Ease (FKRE) and number of certain punctuation characters. Researchers [24] created SpotFake a multimodal framework for fake news detection that used BERT for text feature extraction and showed it outperformed the state-of-the-art model at the time, the model they used for comparison was Event Adversal Neural Network for Multi-Modal Fake News Detection (EANN) by authors in [27]. By using a BERT based model on a cross language fake news detection task in Chinese and English researchers [25] showed it outperformed other methods reaching accuracy of 98% and 96% respectively for each language. Researchers [26] proposed a user preference-aware fake news detection framework UPFD that took into consideration the posts users previously posted to understand their preferences and news engagement on social media. Their framework used

BERT as pretrained embeddings and achieved accuracy of 97.23% on Gossipcop Dataset. An overview [4] of deep learning approaches for fake news detection by researchers] pointed out that the combination of BERT with a 1d-CNN is beneficial for large-scale textual data and that combination can successfully handle ambiguity.

In 2020 there were 750 fake news websites discovered by researchers [28] but without a clear source it is difficult to compare the number of fake news sites today. Researchers compared using a US based general fake news data set and a covid-19 specific fake news dataset and found the latter to perform better with their Bi-LSTM model. They also note that the higher accuracy achieved by a BI-LSTM model is due to its ability to work forward and backward compared to the LSTM which only works forward.

Limited research has been done on the topic of climate chnage fake news, most of the fake news datasets focus on political topics or Covid-19 news articles. There are interesting insights about the climate chnage posts on different social media platforms, such as twitter and facebook, but there seems to be a lack of research about climate change news articles. Because of the reasons mentioned in this section researching the automatic ability of machine leraning models and deep learning models to correclty identify real news and fake news in the domain of climate change in worth time and effort.

2. Models

2.1 Word representations

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure used in text mining to evaluate the importance of a word in a document relative to a collection of documents. The idea behind TF-IDF is that the more frequently a word appears in a document, the more important it is. However, words that appear frequently across many documents, such as common stopwords are less informative, so the inverse document frequency reduces the weight of those common terms. Mathematically, TF-IDF is calculated as the logarithm of the total number of documents divided by the number of documents containing the word. This weighting mechanism effectively reduces the importance of common words and highlights more significant, domain-specific terms. One limitation of TF-IDF is that it does not capture semantic relationships between words, treating terms as independent entities.

Global Vectors for Word Representation (GloVe) is an unsupervised learning algorithm developed by Stanford [63] to generate vector representations of words. It captures semantic meaning from a large corpus of text by aggregating word co-occurrence statistics, allowing the model to identify relationships between words. GloVe's technical structure revolves around the factorization of a word co-occurrence matrix, which captures how often words appear together within a certain window in a large corpus. The model trains by minimizing a least-squares objective function that compares the predicted co-occurrence probabilities to the actual values in the matrix.

Word2Vec is a two-layer neural network model that generates distributed word embeddings by capturing the context of words within a given text corpus. Word2Vec embeddings are powerful because they capture semantic and syntactic relationships between words, such as analogies and word similarity. Technically, Word2Vec relies on two primary architectures: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts the current word based on its surrounding context, while Skip-gram predicts surrounding words from the target word. The model is trained using stochastic gradient descent and leverages negative sampling to reduce the computational cost of calculating softmax across large vocabularies. Word2Vec’s embeddings are dense vectors where similar words are positioned close together in a high-dimensional space, capturing complex semantic relationships.

FastText, developed by Facebook's AI Research (FAIR), is an extension of Word2Vec that incorporates subword information into the word embeddings. Unlike Word2Vec, which treats words as single units, FastText breaks words down into n-grams, enabling the model to represent rare words. This allows the model to generate embeddings for words based on their subword structures, making it particularly effective for morphologically rich languages. FastText’s use of subwords enables it to generalize better to rare or unseen words by building word vectors from smaller components.

2.2 Machine leaning models

Support vector machines (SVMs) are supervised learning models used for classification and regression tasks. The central idea behind SVM is to find a hyperplane that best separates different classes in a high-dimensional space, with the goal of maximizing the margin between the nearest data points of each class, called support vectors. SVMs are effective for high-dimensional spaces and are often used in text classification. They are particularly useful for non-linear classification problems by using kernel functions that map the input data into higher dimensions. SVM employs Lagrange multipliers and solves a quadratic optimization problem to maximize the margin between the classes. For non-linear problems, SVM uses kernel functions like the Radial Basis Function (RBF) or polynomial kernels to map the data into higher-dimensional spaces where a linear separation is possible.

Decision trees (DT) are non-parametric models used for both classification and regression tasks. They work by recursively splitting the data into subsets based on the feature that provides the highest information gain or the lowest Gini impurity. The tree is composed of internal decision nodes which split the data and terminal leaf nodes which represent the final class or value prediction. This greedy algorithm builds the tree from the root, splitting at each node based on the feature that best separates the data. While decision trees are intuitive and interpretable, they can easily overfit to training data, especially when deep trees are constructed. Since they can be prone to overfitting ensemble methods like random forests are often used to improve their generalization performance.

Random forest (RF) is an ensemble learning method primarily used for classification and regression tasks. It operates by constructing a multitude of decision trees during training and outputting the mode of the classes. The method helps reduce the variance and overfitting associated with decision trees by averaging multiple decision trees. Random forest uses techniques like bootstrap aggregation to improve generalization and reduce variance. During the training phase, each tree is grown to its full depth, but overfitting is controlled by averaging the predictions of all the trees.

Extreme gradient boosting (XGBoost) is an advanced implementation of gradient boosting algorithms, designed for efficiency and scalability. It builds multiple decision trees sequentially, where each tree tries to correct the errors of its predecessor. It uses second-order derivatives during optimization, allowing the model to converge faster and more accurately than traditional gradient boosting. XGBoost incorporates several regularization techniques to prevent overfitting, such as L1 and L2 regularization.

Naive Bayes (NB) is a of probabilistic algorithm based on applying Bayes' theorem with the "naive" assumption of conditional independence between features. This assumption simplifies the computation and makes the model highly scalable. In practice, the independence assumption rarely holds, but the model often performs surprisingly well in text classification, where feature vectors tend to be sparse and relatively independent.

Logistic regression (LR) is a statistical model that is used for binary classification problems. It estimates the probability that a given input belongs to a particular class by modeling the relationship between the input features and the output probability using a logistic function. Logistic regression outputs a value between 0 and 1, representing class probabilities. Logistic regression assumes that the relationship between the input features and the log-odds of the class label is linear.

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based model developed by Google [62], designed to pre-train deep bidirectional representations by jointly conditioning on both left and right context in all layers. This approach allows BERT to better understand context of words in a sentence. Unlike traditional models that read text sequentially, BERT is bidirectional, meaning it considers both left and right context simultaneously, allowing it to generate richer word representations. BERT is pretrained on large corpora using masked language modeling (MLM) and next sentence prediction (NSP) tasks, which allow it to learn contextual relationships across sentences.

3. Methodology

3.1 Implementation

In this research, a combination of hardware, programming languages, and software libraries were utilized to achieve the results of the study. Below is a detailed description of the technologies and tools used. The experiments were conducted on two different machines. First a personal laptop was used for initial coding, data preprocessing, and small-scale testing of the models. The specifications of this laptop included an Intel Core i5 processor, 8 GB of RAM, and a 512 GB SSD. Secondly a computer with a RTX 3060TI graphics card was used for more computationally intensive tasks, such as BERT training. The primary programming language used in this research was Python. Python was chosen due to its extensive support for data science and machine learning with multiple libraries and frameworks [3]. For the development of machine learning models, a JupyterHub environment was utilized. This environment was provided by the University of Zagreb University Computing Centre (SRCE), which offers a readily available and pre-configured platform suitable for data science and machine learning tasks. A variety of Python libraries and frameworks were used to handle different aspects of the research, including data manipulation, machine learning model development, and visualization. These libraries are detailed below. The Pandas library was extensively used for data manipulation, including reading and writing data, handling missing values, merging datasets, and performing statistical analyses. NumPy was used alongside Pandas for numerical operations, including array manipulation, mathematical functions, and handling large datasets efficiently. Machine learning models were used form the scikit-learn library. The SVM model was implemented using the sklearn.svm module from the scikit-learn library. The decision tree model was developed using the sklearn.tree module. Linear regression was applied using the sklearn.linear_model module. The Naive Bayes classifier was implemented using the sklearn.naive_bayes module, specifically with the GaussianNB function for handling continuous data. The random forest model was created using the sklearn.ensemble module. To transform text data into numerical features, the TfidfVectorizer from sklearn.feature_extraction.text was utilized. For creating the custom word vectors, Gensim library was used. For natural language processing tasks such as tokenization, lemmatization, and part-of-speech tagging, the Spacy library was employed. Spacy was particularly useful for extracting features based on word type and other linguistic properties. For extracting emotional features from the text data, the LeXmo library was used, which includes a lexicon for identifying and categorizing emotional expressions within the text. To streamline the process of combining textual and numerical features for model training, a pipeline was constructed using the sklearn.pipeline module. This allowed for efficient data preprocessing, feature extraction, and model training and testing in a sequential and repeatable manner. For visualizing the results and generating various plots and graphs, multiple libraries were used. Matplotlib is a versatile library for creating a wide range of static, animated, and interactive visualizations. Built on top of Matplotlib, Seaborn was used to create more aesthetically pleasing and informative statistical graphics. For visual representation of the most frequent terms in the text data, the WordCloud library was employed. These tools and technologies were selected for their robust performance, ease of integration, and widespread use in the data science community. Their collective use facilitated the successful completion of the research objectives, enabling efficient data handling, model training, and result visualization.

3.2 Feature selection

Since textual data has no numerical features it can be beneficial to add certain values based on the text, an example of that would be the use of a sentiment score of a given text. Researchers found sentiment analysis to be a useful technique for evaluating the opinions about climate change [39]. Extracting the topic of a given text was also found beneficial for improving models performance by researchers [40]. Using topic embeddings is a similar approach that improved models performance and added information about the text [41]. References made to non-specific authority were a clear predictor of climate change misinformation in a study on Chinese social media [42]. On the other hand, the same study reported in a scientific content an authority reference conveys trust and expert knowledge. Another informative insight by the same researcher is that true information is more often associated with government sources and misinformation tends to reference non-specific „expert“ sources.

The obtained dataset primarily consisted of one feature – the news article and one label – the fake or real class. The zero or negative label is the real news label while the one or the positive label is the fake news label. The rest of the features added were extracted from the news articles.

Sentiment analysis was shown to be an important feature for improving models accuracies when detecting fake news by researchers [61], in fact they conducted an abation study that showed sentiment was the biggest contributor to a models performance. As sentiment is a frequently added feature to the textual dataset it was also used in our case. To calculate the sentiment of each article by using the TextBlob library that contains a simple API for performing text processing tasks. A study on fake covid news in German concluded that syntactic features were the second most important in improving their models [13]. Similarly in this thesis we used textual features as mean word length in characters, character count and word count. Addition of part of speech (POS) counts was explored in a study by [53] and similarly we included some od those features, such as noun, adjective, verb and named entity counts. Finally emotional values for each article were added, as their importance in fake news detection was highlighted in a study on emotional analysis [45]. To calculate emotional values of each article the LeXmo library was used that contains a lexicon of words and corresponding emotional values for these emotions: anger, anticipation, disgust, fear, joy, sadness, surprise and trust. Since features are added to the dataset by extracting them from the text, it was important to test their impact on the models used. Therefore added features are called „numerical features" through the rest of the thesis.

3.3 Preprocessing

Preprocessing involves many steps to ensure the textual data is clean and well prepared for using in machine learning models. It is important to preprocess raw news articles data because news articles can contain redundant data, such as links or hashtags. Preprocessing involved removing all the non-English words and characters, removing punctuations, links, URLs, hashtags, HTML tags and converting all letters to lowercase. Unstructured data from the dataset had to be removed if it did not fit the comma separated file rules.

3.4 Word representation

To successfully classify text with machine learning models it is necessary to represent the text as accurately as possible in a way that the models can understand it. Word representation can range from simple word indexing to complex word embeddings and vectors. Methods vary in their complexity and their ability to accurately represent and convey the context and meaning of the textual data. Different word representations were used, from term frequency inverse document frequency (TD-IDF) encodings to the complex word vectors. Since TD-IDF is a simpler approach to encoding words it was used as staring point for this thesis. When applying TD-IDF tokenization we can limit the number of tokens used to reduce the dimensionality of the feature space. A different but more accurate representation of words are word embeddings or word vectors. Different word vector building algorithms are used in order to create vector word representations. The ones used in this thesis are GloVe Twitter, Glove Wiki, Word2Vec and FastText embeddings as well as custom embeddings created form the obtained dataset by using the Gensim library and the Word2Vec model was trained on the tokenized sentences from the dataset.

3.5 Model training and testing

The machine learning models used are support vector machines, random forests, naive Bayes, logistical regression, decision trees, a deep neural network, a convolutional neural network and a transformer model BERT. Every model apart from BERT was tested on the following embeddings: TF-IDF, GloVe Twitter 200 dimensions, Glove Wikipedia 300 dimensions, Word2Vec with 100 dimensions, FastText with 100 dimensions and a custom trained Word2Vec model with a 100 dimensions. BERT was not used with the mentioned embeddings, since the model either uses pre-trained embeddings provided by BERT or it fine-tunes the entire model on the specific dataset. Every model was tested with the textual embeddings only and with the addition of numerical features. To gain representative results and understand the models performance the models were first trained on training dataset and then tested on the testing dataset. In the case of a deep neural network, a convolutional neural network and BERT model data was split into three subsets where an addition validation dataset is used for evaluating the model in each epoch of the training process.

The following machine learning models: SVM, RF, NB, LR and DT were all tested in a similar workflow presented below. Firstly the necessary libraries are imported and the dataset is loaded. Secondly the features columns are defined and if additional numerical features were used a ColumnTransformer is used to apply different preprocessing steps to different features creating a preprocessor object. Thirdly, data is split into a training and a test dataset. The fourth step is to create a pipeline with the preprocessor and the classifier model is created. The fifth step is to train the pipeline. The sixth step is to obtain the results and evaluation metrics of the model on the test data. The final step is to generate training accuracy and loss graphs and save results into a text file.

A high level pseudocode for the machine learning models is as follows:

```
1. Import all the necessary libraries, load data
2. Apply preprocessing steps
   IF (only text is used)
         Create a preprocessor object with only textual features
   ELSE (text and numerical features are used)
         Create a preprocessor object with textual and numerical features
3. Split the data into a training and test dataset
4. Create a pipeline with the preprocessor object and the classifier object
5. Train the pipeline
6. Test the pipeline
7.  Save results and generate graphs.
```

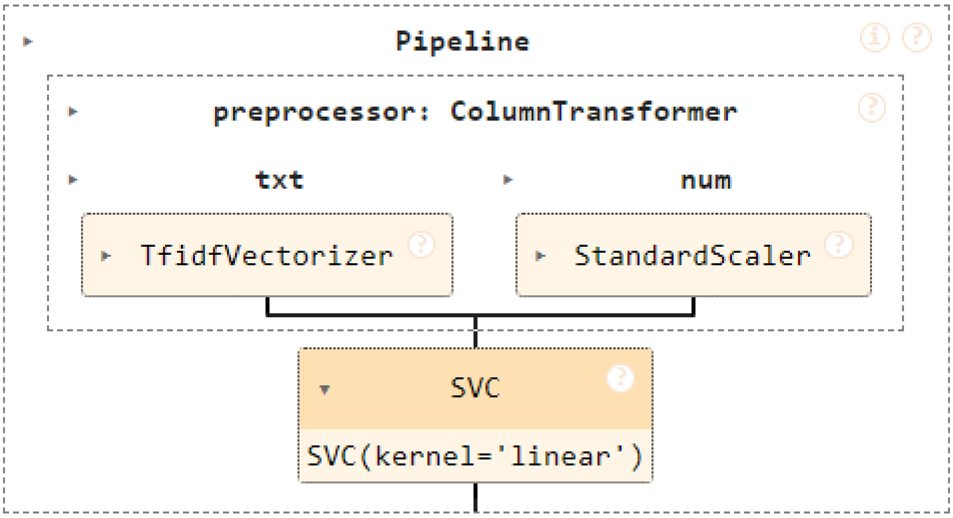

Figure 1. An example of the pipline model for the SVM

On Figure 1. we can see an example of the pipeline for the SVM classifier. First the preprocessor trnsforms text features using the TfIdfVectorizer and scales the numerial features with StandardScaler. These preprocessed features are then passed to the SVM classifier with a linear kernel.

3.6 Hyperparameter optimization

In order to improve accuracy results and models performance hyperparameter optimization was used. Different hyperparameters of different models were tested out by using the grid search approach. This is a straightforward but a time consuming approach because it simply trains the models multiple times with different hyperparameter values based on a grid of values that is being tested. Grid search is one way to search for the optimal parameters, but more specified methods involve using a random search or a genetic algorithm. In this thesis grid search was utilized because it provided a simple way of obtaining improved hyperparameters. The advantage of using grid search is that it is a simple way to explore hyperparameter values but it only searches through the given data and stops there.

4. Results

Results obtained by the methods described in the previous chapter are presenter here. First the task of analysing the dataset is presented, figures with average feature values and distributions are presented to gain better insight into differences and similarities between classes. The features explored in these figures are obtained through analysing the text of the news articles. Secondly the performance values for each of the machine learning model are presented. Models presented in the tables are as follows: SVM, Random Forest, Naive Bayes, Logistic Regression, Decision Trees and XGBoost. For each model there is a table containing it's accuracy, precision, recall and F1-score that was produced on the testing set of datapoints. Each table contains evaluation metrics obtained for different features and word embedding used. Next, in third sub-chapter the best results obtained through hyperparameter optimization for each machine learning model are collected and the best hyperparameters are presented. Lastly the results for different deep learning models are presented in tables for each model. The deep learning models presented are a neural network, a convolutional neural network and BERT.

4.1 Data analysis

This chapter describes different features of the obtained fake news dataset, varying from textual features, linguistical features, stylometry features and numerical features. Figure 2. shows the 20 most frequent words for our fake news dataset.

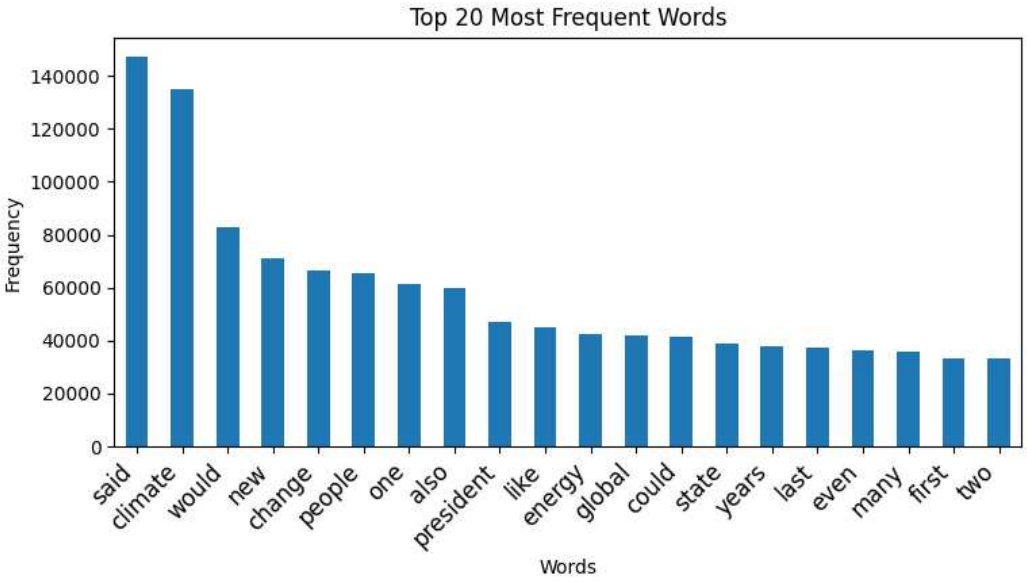

Figure 2. 20 most frequent words in the dataset

Table 1. contains the ten top most common word used in each class. The first column represents the words that were most commonly occurring in the real news class and the third column represents words that were most common in the fake news class. The top ten words from each class differ slightly in their occurrence but we can see that eight of the top ten words overlap in both classes, those words are: „said", „climate", „would", „new", „also", „people", „change" and „one". The similar top ten words is not too surprising since both classes of articles are of a specific topic, in this case climate change.

Table 1. Top ten of the most occurring words in each news articles class

REAL NEWS CLASS		FAKE NEWS CLASS	
WORD	NUMBER OF OCCURRENCES	WORD	NUMBER OF OCCURRENCES
said	122932	*climate*	67005
climate	67928	*would*	35207
would	47792	*change*	32715
new	44343	*people*	28202
also	39049	*one*	28123

people	27380	*new*	26692
change	33691	*said*	24325
one	33265	*global*	23183
state	27621	*also*	20871
water	27008	*like*	20535

Figure 3. shows the class distribution between the fake and real class labeled 1 and 0 respectively. The real class (0) has a greater number of datapoints in the dataset a total of 21491 and the fake class (1) has 17040 datapoints. The entire dataset contains of 38531 news articles.

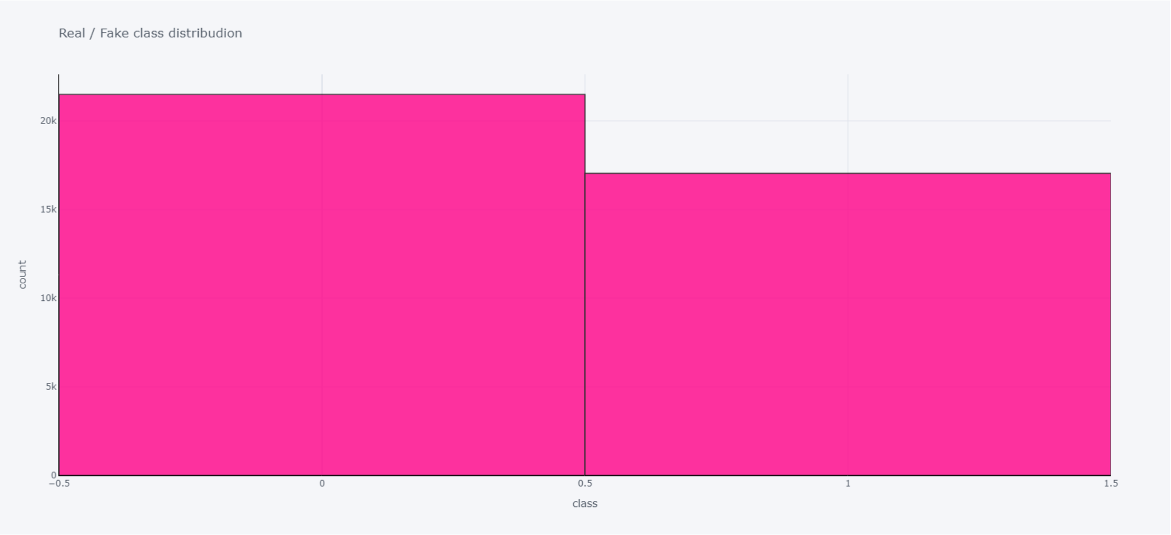

Figure 3. Class distribution (0-real news, 1-fake news)

Figure 4. shows the distribution of sentiment values of the dataset for each class. The real news sentiment is presented in a blue color and the fake news sentiment is colored in organe. It is visible that the distribution graphs for classes mostly overlap. Similar findings are presented in Table 2. regarding the sentiment polarity.

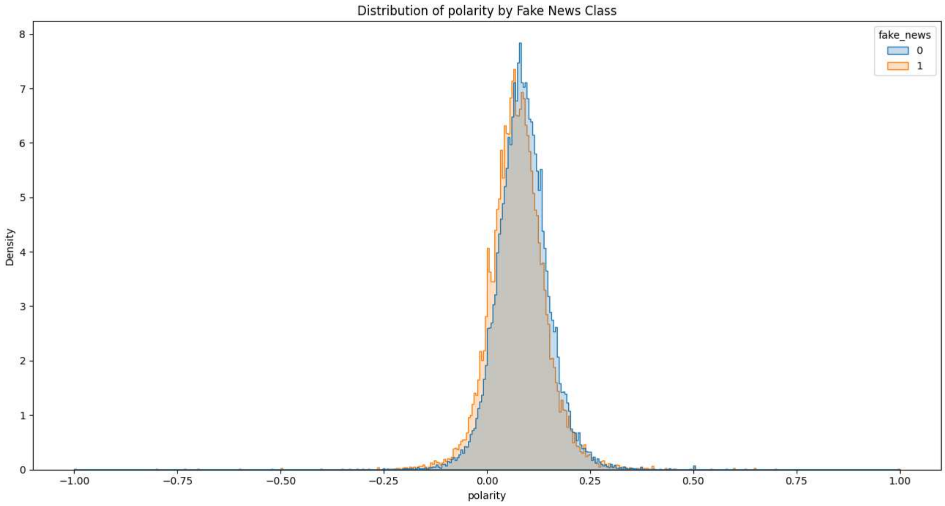

Figure 4. Sentiment Polarity Distribution by each class

In Table 2. the mean values of different features for each class are presented. These mean values give insight into typical characteristics of each class by highlighting the distinguishing features between classes. Sentiment polarity has a similar mean value for real and fake class witch could make it less useful in effectively distinguishing between classes. This is in correspondence to Figure 4. presenting polarity distributions. The sentiment polarity mean values presented in Table 2, are not extreme witch could be due the objectivity of news articles, since it is often a goal aim to write in an objective and informative way in these types of media. Mean values of features character count and word count are presented in Table 2. The character count value is greater for the real news class, and also the word count feature indicating that it could be an informative feature for fake news distinction.

Table 2. Values of features for each class

Feature	**Mean value for Real News Class**	**Mean value for Fake News Class**	**t value**	**p value**
Sentiment Polarity	0.086591	0.073936	18.667	<0.001
Character Count	4794.747103	4072.618134	21.395	<0.001
Word Count	816.273929	700.417958	19.941	<0.001
Mean Word Length	4.892538	4.825446	26.038	<0.001
Named Entity Count	36.180401	27.414847	29.806	<0.001
Noun Count	196.493183	157.223826	28.817	<0.001
Adjective Count	70.005118	60.624296	18.361	<0.001
Verb Count	107.428505	87.480927	26.085	<0.001

Top two graphs in Figure 5. presents the distributions of character count and word count, it is visible that the distributions differ between classes more than the polarity value presented in Figure 4. it could therefore be possible that character count and word count can play a role in discriminating between classes. Figure 6. presents the mean values of character count and word count from the Table 2. and the values correspond to the difference in the distribution of these features.

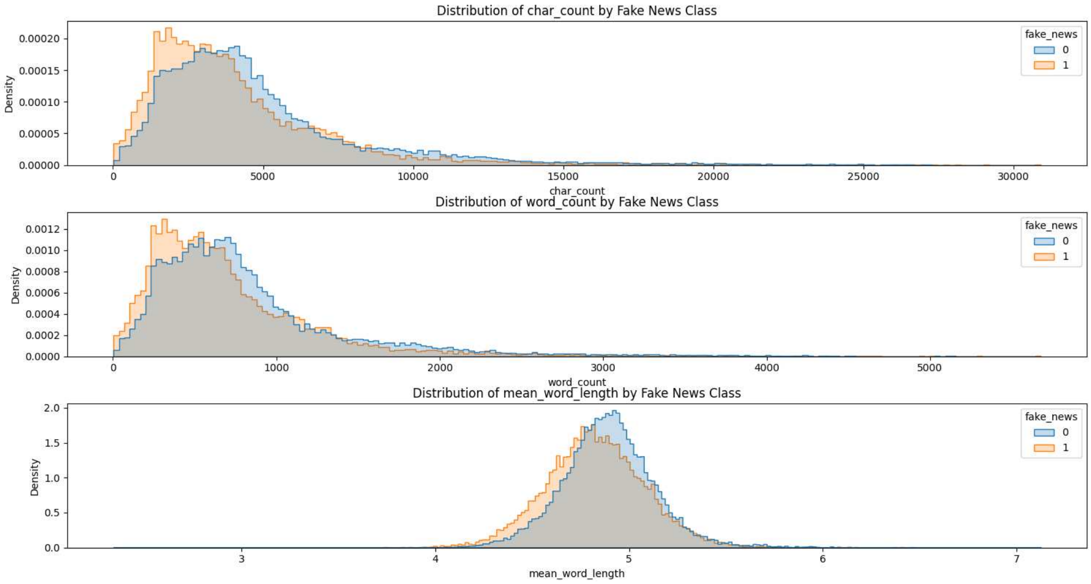

Figure 5. Distribution of numerical textual features by each class

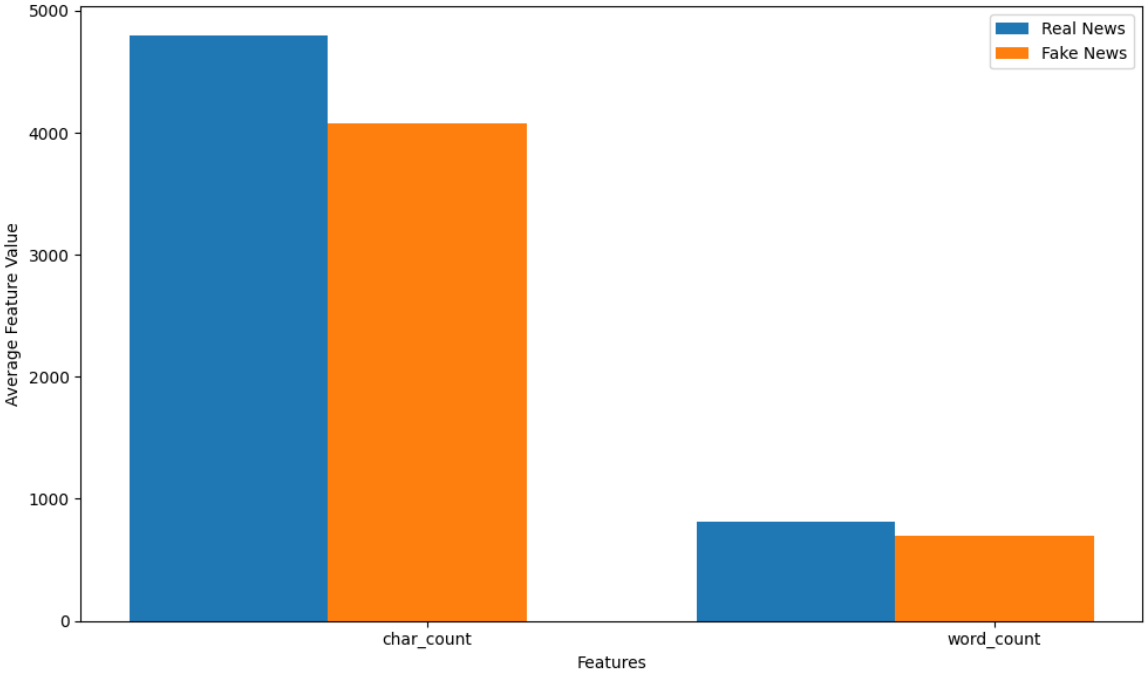

Figure 6. Mean value of numerical features by each class

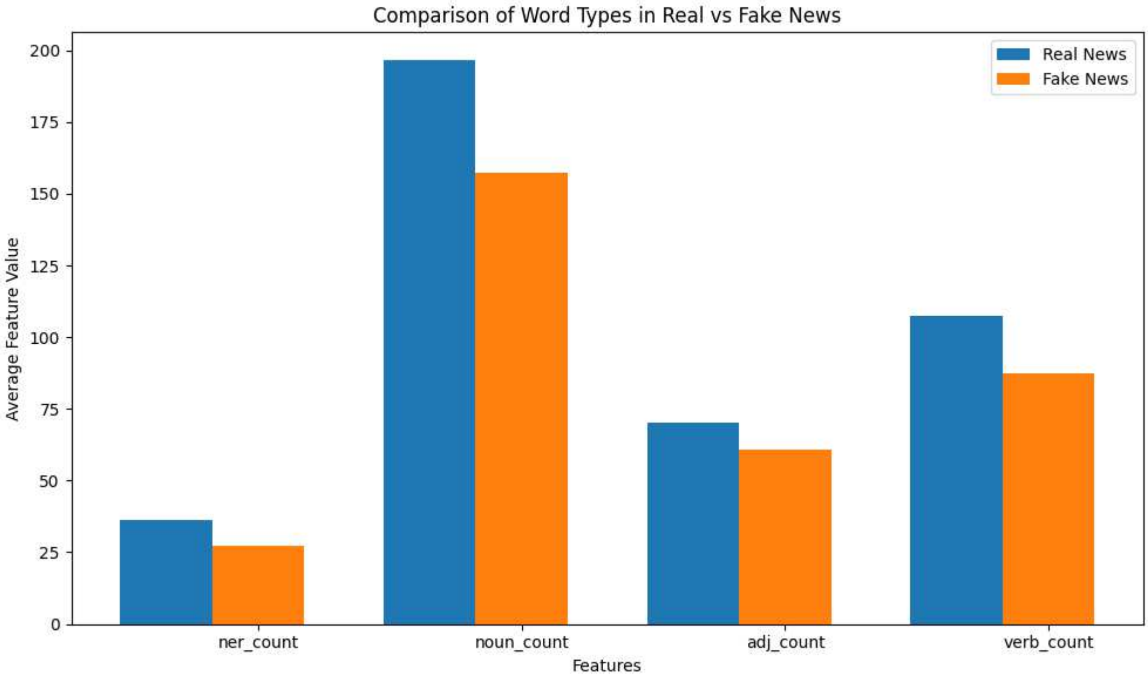

Figure 7. Mean value for word types in each class

Looking at different word types in the text can also give insight about its structure. Figure 7. shows average values of different word types per article. From Table 2. and Figure 7. it is visible that the real class has a greater number of named entities, nouns, adjectives and verbs per article and it can indicate that it is an informative feature of that class. Figure 8 shows distributions of these word type features by each class.

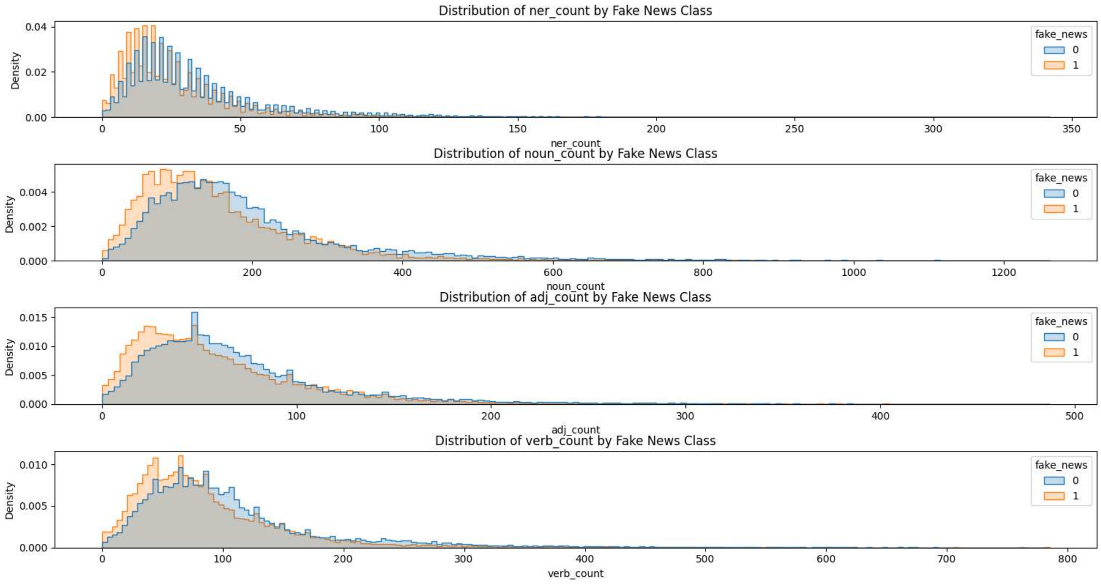

Figure 8. Distribution of word type features by each class.

Table 3. shows the mean values for emotions in each class. Mean emotional values are greater for the fake news class in the following emotions: anger, disgust, fear, joy, sadness and surprise. For the real news class only the mean emotional values of anticipation and trust are higher than the fake news class. This is also visible in Figure 9. as a bar chart. Emotional values are calculated by the help of an emotional lexicon and the values for each of the emotions ranges from [-1,+1].

Table 3. Mean values for each emotion in each class

Emotion	**Mean Value for Real News Class**	**Mean Value for Fake News Class**	**t value**	**p value**
anger	0.010933	0.012980	-22.755	<0.001

anticipation	0.024835	0.023256	15.148	<0.001
disgust	0.004834	0.006390	-26.937	<0.001
fear	0.014235	0.016054	-16.314	<0.001
joy	0.013214	0.014112	-9.633	<0.001
sadness	0.010284	0.011566	-15.909	<0.001
surprise	0.009303	0.009918	-8.360	<0.001
trust	0.034682	0.033511	8.668	<0.001

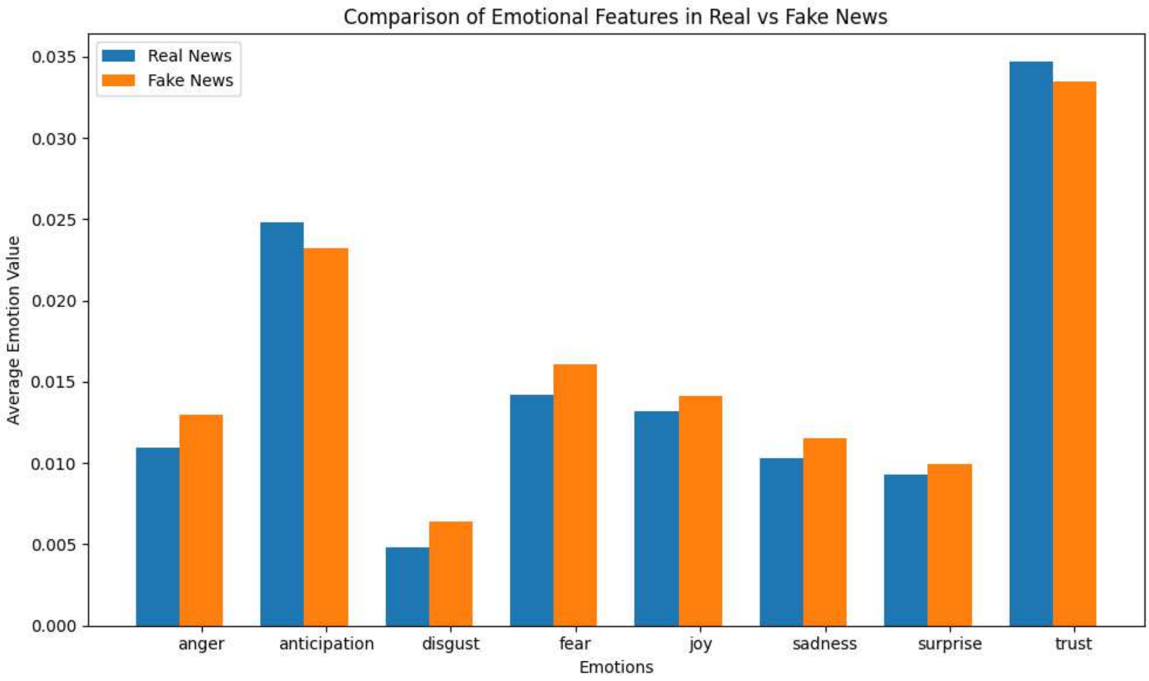

Figure 9. Mean emotional values by each class

4.2 Machine learning models

In this section the results from each of the machine learning models are presented. Every model is evaluated with accuracy, precision, recall and an F1-score. Models were tested using different embedding types and features. The best combination of features and embeddings is bolded for each model.

Table 4. presents the results obtained by using support vector machines for the fake news classification task. The SVM model achieved the highest overall accuracy (0.9033) when using TF-IDF embeddings combined with text and numerical features with strong F1-scores for both classes. TF-IDF embeddings outperforms other embedding types with the model maintaining similar precision and recall values wether numerical features were included or not. GloVe embeddings trained on Wikipedia data with 300 dimensions provided better results than those trained on twitter data and 200 dimensions. The FastText embedding scored the lowest accuracy (0.7850) indicating that this embedding may be less effective for the fake news articles classification task. Values for accuracy, precision, recall and F1-score are similar when using only textual features as compared to using textual and numerical features and in some cases, such as FastText, Word2Vec and Custom100d, the results are of a smaller value when using the numerical features

Table 4. SVM model

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	Text only	0.9013	0.92	0.88	0.90	0.90	0.91	0.89
TF-IDF	**Text + numerical features**	**0.9033**	**0.92**	**0.88**	**0.90**	**0.90**	**0.91**	**0.89**
GloVe twitter 200d	Text only	0.8198	0.84	0.80	0.84	0.79	0.84	0.79
GloVe twitter 200d	Text + numerical features	0.8175	0.84	0.79	0.84	0.79	0.84	0.79

GloVe wiki 300d	Text only	0.8462	0.86	0.82	0.86	0.83	0.86	0.82
GloVe wiki 300d	Text + numerical features	0.8475	0.87	0.82	0.86	0.83	0.86	0.83
FastText	Text only	0.8081	0.82	0.79	0.85	0.76	0.83	0.78
FastText	Text + numerical features	0.7850	0.80	0.77	0.83	0.73	0.81	0.75
Word2Vec	Text only	0.8426	0.86	0.82	0.86	0.82	0.86	0.82
Word2Vec	Text + numerical features	0.8335	0.85	0.81	0.85	0.81	0.85	0.81
Custom100dVec	Text only	0.8111	0.83	0.79	0.83	0.78	0.83	0.78
Custom100dVec	Text + numerical features	0.7850	0.80	0.77	0.83	0.73	0.81	0.75

Table 5. presents the results obtained by using random forests for the fake news classification task. The RF model achieved the highest overall accuracy (0.8896) when using TF-IDF embeddings combined with text and numerical features with strong F1-scores for both classes. TF-IDF embedding outperforms other embedding types. GloVe embeddings trained on Wikipedia data with 300 dimensions provided better results than those trained on twitter data and 200 dimensions. The Glove twitter 200 dimensional embedding scored the lowest accuracy (0.8000) indicating that this embedding may be less effective for the fake news articles classification task. The inclusion of numerical features consistently enhanced the performance of the RF model across all embedding types, particularly for models using FastText and Word2Vec embeddings.

Table 5. Random Forest model

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	Text only	0.8747	0.88	0.87	0.9	0.84	0.89	0.85
TF-IDF	**Text + numerical features**	**0.8896**	**0.90**	**0.87**	**0.90**	**0.88**	**0.90**	**0.87**
GloVe twitter 200d	Text only	0.8000	0.81	0.79	0.85	0.74	0.83	0.76
GloVe twitter 200d	Text + numerical features	0.8094	0.82	0.80	0.85	0.76	0.83	0.78
GloVe wiki 300d	Text only	0.8117	0.81	0.81	0.86	0.75	0.84	0.78
GloVe wiki 300d	Text + numerical features	0.8216	0.83	0.81	0.86	0.77	0.84	0.79
FastText	Text only	0.8103	0.82	0.80	0.85	0.76	0.83	0.78
FastText	Text + numerical features	0.8177	0.83	0.81	0.85	0.77	0.84	0.79
Word2Vec	Text only	0.8128	0.82	0.81	0.86	0.75	0.84	0.78
Word2Vec	Text + numerical features	0.8179	0.82	0.81	0.86	0.76	0.84	0.79
Custom100dVec	Text only	0.8137	0.82	0.80	0.85	0.76	0.84	0.78
Custom100dVec	Text + numerical features	0.8206	0.83	0.81	0.86	0.77	0.84	0.79

Table 6. presents the results from testing the Naive Bayes model on a fake news classification task. The NB model archived the highest accuracy (0.8454) when using only textual features and TF-IDF embeddings, making it the most effective combination in this context. Numerical features to the TF-IDF embeddings decreased the accuracy values, implying it has minimal impact on the model's performance. Other embedding types and feature combination yielded results of accuracy between 0.7054 and 0.7336, witch could indicate that those combinations are not suitable for our task.

Table 6. Naive Bayes model

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	**Text only**	**0.8454**	**0.87**	**0.81**	**0.85**	**0.84**	**0.86**	**0.83**
TF-IDF	Text + numerical features	0.8436	0.87	0.81	0.85	0.83	0.86	0.82
GloVe twitter 200d	Text only	0.7110	0.76	0.66	0.72	0.70	0.74	0.68
GloVe twitter 200d	Text + numerical features	0.7205	0.76	0.67	0.73	0.71	0.75	0.69
GloVe wiki 300d	Text only	0.7054	0.76	0.65	0.69	0.72	0.73	0.68
GloVe wiki 300d	Text + numerical features	0.7137	0.77	0.66	0.70	0.73	0.73	0.69
FastText	Text only	0.7324	0.76	0.69	0.76	0.70	0.76	0.70
FastText	Text + numerical features	0.7336	0.76	0.70	0.76	0.70	0.76	0.70
Word2Vec	Text only	0.7087	0.77	0.65	0.69	0.73	0.73	0.69
Word2Vec	Text + numerical	0.7172	0.77	0.66	0.71	0.73	0.74	0.69

	features							
Custom100dVec	Text only	0.7110	0.76	0.66	0.71	0.72	0.73	0.68
Custom100dVec	Text + num features	0.7226	0.77	0.67	0.72	0.72	0.74	0.70

The Logistic Regression model's results are presented in Table 7. and the highest accuracy obtained was 0.9072 by using TF-IDF embeddings with text-only features, with high values for precision and recall across both classes. Adding numerical features to the TF-IDF embeddings resulted in a slight decrease in accuracy to 0.9064, with minimal changes in precision, recall, and F1-scores. The GloVe embeddings trained on Wikipedia with 300 dimensions performed better than the Twitter-trained embeddings, with an accuracy of 0.8401 for text-only features, but saw a slight drop when numerical features were added. FastText embeddings produced the lowest accuracy (0.7587) and showed weaker performance compared to TF-IDF and GloVe embeddings. Custom embeddings with 100 dimensions performed similarly to GloVe Twitter embeddings, with slight improvements when numerical features were included, but overall, they were bested by the TF-IDF-based approach.

Table 7. Logistic reggresion model

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	**Text only**	**0.9072**	**0.92**	**0.89**	**0.91**	**0.90**	**0.92**	**0.89**
TF-IDF	Text + numerical features	0.9064	0.92	0.89	0.91	0.90	0.92	0.89
GloVe twitter 200d	Text only	0.8030	0.82	0.78	0.83	0.76	0.83	0.77
GloVe twitter 200d	Text + numerical features	0.8021	0.82	0.78	0.83	0.77	0.82	0.77
GloVe wiki	Text only	0.8401	0.86	0.84	0.86	0.81	0.86	0.82

300d								
GloVe wiki 300d	Text + numerical features	0.8337	0.85	0.81	0.85	0.81	0.85	0.81
FastText	Text only	0.7587	0.76	0.76	0.84	0.66	0.80	0.71
FastText	Text + numerical features	0.7593	0.77	0.75	0.82	0.68	0.79	0.71
Word2Vec	Text only	0.8175	0.83	0.8	0.85	0.77	0.84	0.79
Word2Vec	Text + numerical features	0.8153	0.83	0.80	0.85	0.77	0.84	0.79
Custom100dVec	Text only	0.8085	0.82	0.79	0.84	0.77	0.83	0.78
Custom100dVec	Text + numerical features	0.8118	0.83	0.79	0.84	0.78	0.83	0.78

Table 8. presents results for the decision tree model. The highest accuracy obtained is 0.8132 by using TF-IDF embeddings combined with text and numerical features. Both TF-IDF configurations text-only and text with numerical features produced similar performance results. GloVe embeddings trained on Twitter (200d) resulted in the lowest accuracy 0.6939, with precision, recall, and F1 scores consistently lower compared to TF-IDF embeddings. The addition of numerical features to GloVe embeddings led to slight improvements in accuracy and F1 scores, though the overall performance remained below that of TF-IDF embeddings. FastText and Word2Vec embeddings had similar performance, with accuracy value of approximately 0.7100 and stable precision and recall values. Custom embeddings with 100 dimensions showed modest performance, with slight gains when numerical features were included, but overall, they were bested by the TF-IDF-based approach in all metrics.

Table 8. Decisiton Tree model

Embedding	Feature type	Accuracy	Precision	Recall	F1 Score

type			**0**	**1**	**0**	**1**	**0**	**1**
TF-IDF	Text only	0.8123	0.84	0.78	0.83	0.79	0.83	0.79
TF-IDF	**Text + numerical features**	**0.8132**	**0.84**	**0.78**	**0.83**	**0.79**	**0.83**	**0.79**
GloVe twitter 200d	Text only	0.6939	0.73	0.65	0.73	0.65	0.73	0.65
GloVe twitter 200d	Text + numerical features	0.6947	0.73	0.65	0.73	0.65	0.73	0.65
GloVe wiki 300d	Text only	0.7067	0.74	0.67	0.74	0.66	0.74	0.66
GloVe wiki 300d	Text + numerical features	0.7114	0.74	0.67	0.75	0.67	0.74	0.67
FastText	Text only	0.7101	0.74	0.67	0.74	0.67	0.74	0.67
FastText	Text + numerical features	0.7076	0.74	0.67	0.74	0.67	0.74	0.67
Word2Vec	Text only	0.7042	0.74	0.66	0.74	0.66	0.74	0.66
Word2Vec	Text + numerical features	0.7048	0.74	0.66	0.74	0.66	0.74	0.66
Custom100dVec	Text only	0.7039	0.74	0.66	0.74	0.66	0.74	0.66
Custom100dVec	Text + numerical features	0.7123	0.75	0.67	0.74	0.68	0.74	0.67

Table 9. contains results obtained by testing the XGBoost model on the fake news classification task. XGBoost model achieved its highest accuracy of 0.9200 when using TF-IDF embeddings combined with text and numerical features, showing strong precision (0.94) and recall (0.92) across both classes. The inclusion of numerical features with TF-IDF embeddings slightly improved the accuracy and recall, indicating that these additional

features contributed positively to the model's performance. All other embedding types apart from TF-IDF scored very similarly across all metrics. GloVe embeddings trained on Twitter with 200 dimensions and Wikipedia with 300 dimensions showed high accuracy scores, with the Wikipedia trained embeddings slightly outperforming the Twitter trained embeddings. Across all embedding types, the addition of numerical features generally led to small improvements in the model's performance metrics, in recall and F1 score. TF-IDF embeddings with additional numerical features outperformed all other embedding types, making them the most effective combination for the XGBoost model in the fake news classification task.

Table 9. XGBoost model results

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	Text only	0.9160	0.94	0.89	0.91	0.92	0.92	0.91
TF-IDF	**Text + num features**	**0.9200**	**0.94**	**0.89**	**0.92**	**0.93**	**0.93**	**0.91**
GloVe twitter 200d	Text only	0.85	0.86	0.83	0.87	0.82	0.86	0.83
GloVe twitter 200d	Text + num features	0.86	0.87	0.83	0.87	0.84	0.87	0.84
GloVe wiki 300d	Text only	0.87	0.89	0.85	0.89	0.85	0.89	0.85
GloVe wiki 300d	Text + num features	0.87	0.89	0.86	0.89	0.86	0.89	0.86
FastText	Text only	0.87	0.88	0.85	0.89	0.84	0.88	0.85
FastText	Text + num features	0.87	0.88	0.85	0.88	0.85	0.88	0.85
Word2Vec	Text only	0.87	0.88	0.85	0.89	0.85	0.88	0.85
Word2Vec	Text + num features	0.87	0.89	0.85	0.89	0.86	0.89	0.86

Custom100dVec	Text only	0.84	0.86	0.83	0.87	0.82	0.86	0.82
Custom100dVec	Text + num features	0.85	0.87	0.84	0.87	0.83	0.87	0.83

Table 10. contains the best results for each model (SVM, RF, NB, LR, DT and XGBoost) obtained from Table 4, Table 5, Table 6, Table 7, Table 8 and Table 9. Table 10. shows that the XGBoost model achieved the highest accuracy of 0.9200 using TF-IDF embeddings combined with text and numerical features, demonstrating superior performance across all metrics when comparing to other models. The Logistic Regression model also obtained high accuracy of 0.9072, closely followed by the SVM model at 0.9033, both showing strong precision and recall values when using TF-IDF embeddings. The Random Forest model had a slightly lower accuracy of 0.8896, while the Naive Bayes model underperformed with an accuracy of 0.8454 using TF-IDF text-only features. The Decision Tree model showed the lowest accuracy among the models (0.8132) when using TF-IDF embeddings combined with numerical features, indicating that it was less effective for this classification task.

Table 10. Best results from each ML model

Model	**Embedding type**	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
				0	**1**	**0**	**1**	**0**	**1**
SVM	TF-IDF	Text + numerical features	0.9033	0.92	0.88	0.90	0.90	0.91	0.89
RF	TF-IDF	Text + numerical features	0.8896	0.90	0.87	0.90	0.88	0.90	0.87
NB	TF-IDF	Text only	0.8454	0.87	0.81	0.85	0.84	0.86	0.83
LR	TF-IDF	Text only	0.9072	0.92	0.89	0.91	0.90	0.92.	0.89
DT	TF-IDF	Text + numerical	0.8132	0.84	0.78	0.83	0.79	0.83	0.79

		features							
XGBoost	**TF-IDF**	**Text + numerical features**	**0.9200**	**0.94**	**0.89**	**0.92**	**0.93**	**0.93**	**0.91**

4.3 Hyperparameter optimization

Table 11. Hyperparameter optimisation results for machine learning models

Model	**Embedding type**	**Feature type**	**Parameters**	**Accuracy**
SVM	TF-IDF	Text only	C: 10 Gamma: scale Kernel: rbf TF-IDF max features: 10000 TF-IDF ngram range: (1,2)	0.92
RF	TF-IDF	Text + numerical features	Max depth: 40 Max features : sqrt Min samples leaf: 1 Min saples split: 2 N estimators: 400	0.89

4.4 Deep learning models

This section shows the results from each of the deep learning models. Every model is evaluated with accuracy, precision, recall and an F1-score. Models were tested using different embedding types and features. The best combination of features and embeddings is bolded for each model in the tables.

Table 12. contains the different evaluation metrics for the neural network on the fake news classification task. This deep neural network model achieved its highest accuracy of 0.9000 using TF-IDF embeddings with text-only features, demonstrating high value for precision and balanced F1 scores for both classes.

GloVe embeddings trained on Wikipedia with 300 dimensions obtained slightly better results than those obtained by trained with Twitter embeddings with 200 dimensions, with an accuracy of 0.87 when combined with numerical features, showing consistent recall and F1 scores across both classes. FastText embeddings produced moderate results, though the evaluation metrics slightly decreased to when numerical features were added, particularly affecting recall and F1 scores for the positive class. Word2Vec and Custom embeddings yielded similar results, that can be seen as high, with an accuracy of 0.85-0.86 with text-only features, and maintaining strong performance with slight improvements when numerical features were included.

Table 12. Deep neural network results

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	**Text only**	**0.90**	**0.93**	**0.87**	**0.89**	**0.91**	**0.91**	**0.89**
TF-IDF	Text + num features							
GloVe twitter 200d	Text only	0.84	0.86	0.81	0.85	0.82	0.86	0.81
GloVe twitter 200d	Text + num features	0.84	0.86	0.81	0.85	0.82	0.86	0.81
GloVe wiki	Text only	0.86	0.87	0.85	0.89	0.82	0.88	0.84

300d								
GloVe wiki 300d	Text + num features	0.87	0.89	0.85	0.88	0.85	0.88	0.85
FastText	Text only	0.84	0.86	0.82	0.86	0.81	0.86	0.81
FastText	Text + num features	0.82	0.81	0.84	0.89	0.73	0.85	0.78
Word2Vec	Text only	0.86	0.89	0.83	0.86	0.87	0.88	0.85
Word2Vec	Text + num features	0.85	0.88	0.82	0.85	0.85	0.87	0.84
Custom100dVec	Text only	0.85	0.87	0.81	0.85	0.84	0.86	083
Custom100dVec	Text + num features	0.86	0.88	0.84	0.88	0.84	0.88	0.84

Table 13. contains results obtained from training and testing a convolutional neural network. The CNN model achieved its highest accuracy of 0.90 using TF-IDF embeddings without numerical features, showing a balanced F1 scores for both classes. GloVe embeddings, particularly those trained on Twitter with 200 dimensions, performed poorly, with accuracy as low as 0.57 for text-only features, indicating significant challenges in effectively classifying fake news with these embeddings. Word2Vec and Custom100dVec embeddings produced moderate results, with the highest accuracy of 0.75 and 0.76, respectively, when using text-only features, however adding numerical features lowered the values of the evaluational metrics. The inclusion of numerical features generally did not enhance the performance of the CNN across most embeddings, with some configurations even seeing a decrease in accuracy and F1 scores, particularly with FastText and GloVe Wikipedia embeddings.

Table 13. Convolutional neural network results

Embedding type	**Feature type**	**Accuracy**	**Precision**		**Recall**		**F1 Score**	
			0	**1**	**0**	**1**	**0**	**1**
TF-IDF	**Text only**	**0.90**	**0.94**	**0.85**	**0.88**	**0.93**	**0.91**	**0.89**
TF-IDF	Text + num features	0.90	0.93	0.87	0.89	0.92	0.91	0.89
GloVe twitter 200d	Text only	0.57	0.57	0.00	1.00	0.00	0.72	0.00
GloVe twitter 200d	Text + num features	0.63	0.61	0.68	0.90	0.26	0.73	0.38
GloVe wiki 300d	Text only	0.72	0.72	0.73	0.83	0.57	0.77	0.64
GloVe wiki 300d	Text + num features	0.62	0.62	0.62	0.85	0.33	0.72	0.43
FastText	Text only	0.69	0.68	0.72	0.86	0.47	0.76	0.57
FastText	Text + num features	0.63	063	0.62	0.82	0.38	0.71	0.47
Word2Vec	Text only	0.75	0.77	0.73	0.81	0.68	0.79	0.70
Word2Vec	Text + num features	0.63	0.67	0.57	0.68	0.55	0.67	0.56
Custom100dVec	Text only	0.76	0.81	0.70	0.74	0.77	0.77	073
Custom100dVec	Text + num features	0.69	0.70	0.66	0.77	0.58	0.74	0.62

Table 14. BERT results

Epochs	**Learning rate**	**Dropout**	**Accuracy**	**Loss**	**Precision**	**Recall**	**F1 score**
10	5e-5	0.1	0.9003	0.4000	0.8899	0.8818	0.8858
10	5e-5	0.3	0.9084	**0.3683**	0.8716	**0.9275**	0.8986
10	5e-5	0.5	0.9092	0.4670	0.8945	0.8989	0.8967
10	1e-5	0.1	0.9184	0.4257	0.8976	0.9187	0.9080
10	1e-5	0.3	0.9208	0.3898	0.9132	0.9053	**0.9092**
10	1e-5	0.5	**0.9218**	0.4181	**0.9277**	0.8910	0.9089

The BERT model was trained for 10 epoch under differently parameters of learning rate and dropout. With a learning rate of 1e-5 and a dropout value of 0.5 the BERT model achieved the highest accuracy of 0.9218 and highest precision value of 0.9277. The increase of the dropout value consistently improved the accuracy regardless of the learning rate, but some of the loss values did grow with higher dropout values. Such is the case for both learning rates tested, where a dropout rate od 0.3 produced the lowest loss scores.

5. Discussion

The results presented in the previous chapter show a comprehensive analysis of the evaluation metrics of various machine learning models and deep learning models for the fake news classification task. Firstly, the dataset was explored through various feature analysis techniques, including the visualization of word frequencies and sentiment polarity distributions. The analysis revealed that both real and fake news articles share the most common words, which is not surprising since they both the specific topic of climate change. However, differences in features such as character count, word count, and certain word types and emotional values suggested potential indicators for distinguishing between the two classes. Similar research found that adding emotional features improved the accuracy of models, and authors point out that detecting fake news articles was significantly improved with emotional features [45]. Looking more closely at the emotional analysis of our data set we can see that most emotions score a higher value for the fake news articles, and one of those emotions is anger which was also the case for researchers when looking at emotional values of fake and real news articles [46]. All eight of the emotional values of the dataset were found to be statistically significant with p-values well below the threshold of 0.001. This finding is also in lin with the work presented by researchers [46] where they found similar p-values for the emotional features when comparing the real and the fake class of their English dataset. Whilst it is statistically significant, having very low p-values does not mean these features play a significant role when used with different classifiers, some models even performed worse when additional features were added, such as the deep neural network. Having a large sample size can cause the p-values to be very low, witch is true for this case. The mean values for the emotional features are relatively small since the emotional dictionary contains values from [-1,+1]. This could be due to the higher number of words in each article, diluting the emotional impact. Another reason for the lower emotional values could be that they are written objectively.

The machine learning models tested included SVM, Random Forest, Naive Bayes, Logistic Regression, Decision Trees, and XGBoost.

The SVM model using TF-IDF embeddings combined with text and numerical features achieved a high F1 score of 0.91, showcasing the effectiveness of TF-IDF in capturing relevant textual patterns. Comparing with similar work [23], [47] the obtained results show improvement in F1 score, the mentioned previous work achieved an F1 score of 0.58 and 0.34 respectively with the SVM model on a climate change fake news dataset with additional features such as sentiment and linguistic features. The reason for such improvement in our results for the SVM model could be due to the size of the dataset, given that the SVM model benefits from using a large dataset [48]. Other fake news detection work has also found SVM to be a successful model to use with political and covid-19 datasets both achieving F1 scores of 0.93 [7], [49].

The random forest model obtained an accuracy of 0.88 with the use of tf-idf and additional features, witch is not better than the work presented by researchers [50], where they obtained an accuracy of 0.91 with the random forests model on a political fake news dataset. While our result is not an improvement it is still a viable model for successful fake news detection since we used a climate change focused dataset. Similar to our results adding multiple features to the text improved evaluation metrics for the random forest model as shown by researchers [51]. Another study also achieved high accuracy scores of 0.97 for the random forest model [52], however these results are not readily comparable with ours since they were obtained on a dataset of fraudulent job applications, and not a news articles dataset. Similar to the SVM model, many random forest models were tested on political and covid-19 fake news dataset and research shows promising results, some studies even obtaining accuracies as high as 0.96 for this task [53].

The multinomial naive Bayes model achieved the highest score of accuracy and other evaluation metrics when used with td-idf representation and no additional features, having the F1 score of 0.86 for the negative class and 0.83 for the positive class. This result is an improvement when compared to other climate change classification results by multinomial naive Bayes, where researchers obtained 0.77 and 0.36 respectively [23], [47]. In our results there is a small decline of the evaluation metrics values when additional features added, whereas adding a sentiment value to the text increased the F1 score obtained by researchers [47]. One positive side of using a Naive Bayes classifier is that it can work well even with small datasets, as shown by researchers [48], where they obtained an accuracy of 0.85 when using td-idf representation on a political fake news dataset. Another study looked at the naive Bayes model for fake news classification on a political dataset and obtained an impressive accuracy of 0.97 [52].

The logistic regression model obtained the highest accuracy od 0.90 and other high evaluation metric when used only with tf-idf representations with no additional features. Similar results were obtained by other researchers, however their datasets contained political fake news articles. This study shows a very similar result with an accuracy of 0.91 [48] when tf-idf representations were used. Our results for the logistic regression model with the Word2Vec 300 dimensional embeddings has an accuracy value of 0.84, and it outperforms a similar model with the same embedding used for a logistic regression model where researchers obtained an accuracy of 0.72 [45]. However other research produced a better result than ours for this task, researchers used term frequencies for the text representation and obtained a high accuracy of 0.96 with a logistic regression model on a politics fake news dataset, since their text representations methods are not described in detail it is less clear how to reproduce such a high result [7]. Similar results to ours were obtained for two different studies on a covid-19 dataset, where both researchers had an accuracy of 0.91 for the logistic regression model [49], [53].

The decision trees obtained an accuracy of 0.81 with textual and additional linguistic features. These results are not an improvement when we compare them to similar studies, however there was no comparison with a similar climate change themed dataset, most studies focused on political fake news and used decision trees, such as [56] where researchers obtained an accuracy of 0.99 and 0.90 for different datasets using tf-idf. Similar results were also achieved by researchers in [53] where different features were tested, their results show only a small difference in accuracy and other evaluation metrics with or without additional features, with accuracies between 0.92 and 0.93, a similar effect was shown in our results, where additional features did not impact the models performance greatly. Another study used decision trees for fake news detection and obtained an accuracy of 0.88 with identical hyperparameter values as our model [57], witch could point to the differences in data. A similar effect was shown in a study where multiple models were tested on two datasets and the evaluation metrics varied significantly between the datasets, one of the models used was a decision tree and the accuracy on the FA-KES dataset was 0.55 while the ISOT dataset produced a model with an accuracy of 0.96 [12]. Even though our best performing combination for the decision tree was a tf-idf representation with additional features, our results for the decision tree with GloVe wiki embeddings outperform the results shown in [22] with the same embeddings used.

The XGBoost model outperforms all other models with an accuracy of 0.92, highlighting its superior ability to leverage both textual and numerical features. This result outperforms results obtained by diferent studies where the XGBoost model had lower accuracy scores [10], [59], [60]. The reasons for improvement could be different hyperparameters of the model since no research presented has noted their values, also these studies all used politically themed fake news dataset, making it hard to truly interpret the differences in results. A similar result to ours was obtained by researches [58] when using an all-discourse approach on the Politifact dataset. GloVe embeddings, particularly those trained on Wikipedia with 300 dimensions, consistently provided better results than their Twitter trained counterparts. This is likely due to the more generalizable and context-rich nature of Wikipedia trained embeddings, which could be similar to the news articles. When analyzing the results of the deep learning models, the neural network and CNN models both achieved their highest accuracy of 0.90 using TF-IDF embeddings with text-only features. When looking at similar research on climate change datasets we can see our accuracy is higher than that presented in [47], where a deep neural network with CountVector embeddings had a 0.44 accuracy value. Our results reinforce the conclusion that TF-IDF remains a strong baseline embedding technique, even in the context of deep learning models, somewhat higher results were shown in [22] where researchers obtained an accuracy of 0.94 by using TF-IDF and a neural network. However, the addition of numerical features did not significantly improve the performance of these models, and in some cases, it even resulted in decreased accuracy and F1 scores, particularly for FastText and GloVe embeddings. This could suggest that the numerical features used were either redundant or not sufficiently informative in the presence of textual embeddings. Interestingly, the CNN model struggled with GloVe embeddings trained on Twitter, achieving an accuracy as low as 0.57, which indicates significant challenges in effectively classifying fake news with these embeddings. This may be due to the specific characteristics of the Twitter trained embeddings, which are more suited to short and informal text, whereas the dataset in this study consisted of longer formal news articles. Similar research has obtained a much higher accuracy value of 0.91 for GloVe embeddings with a CNN [22], however it is unclear weather they used Twitter trained or Wikipedia trained GloVe embeddings.

Our results obtained on the BERT model showed the highest accuracy of 0.9218. Different learning rates and dropout values were tested, and lower learning rates proved better for training as well as higher dropout rates. This could be due to the BERT model fine tuning the embeddings on the given dataset, since lower learning rates allow the model to slowly learn on the new dataset and higher dropout values improve generalisation and prevent overfitting. When comparing to a similar study on French fake news about climate change [23] with 6 hand crafted features similar to ours, researchers obtained an F1 score of 0.8475, our results show a higher F1 score of 0.9092. Similarly their BERT model outperforms other machine learning models, witch is also true for our results.

Overall, the results suggest that while traditional machine learning models like SVM and ensemble methods like XGBoost perform exceptionally well with TF-IDF embeddings, deep learning models may require more sophisticated or domain-specific embeddings to reach similar levels of performance. The performance gap observed between different embedding types and models indicates that the choice of embedding and feature set is crucial for the success of fake news classification tasks. These findings contribute to the ongoing research on the application of machine learning and deep learning in misinformation detection, specifically in the climate change domain, and future work could explore the integration of more advanced embedding techniques, such as contextual embeddings from transformers, to further improve classification accuracy.

Shortcomings of this research are multiple, and they could be avoided with these suggested improvements. Integrating metadata about the articles, such as user interactions, post charachteristic and publisher information could be crucial, researchers used this additional information to build accurate models for early fake news detection [10], [11]. Since articles only include the text additional context about the article itself could be beneficial. Adding more sophisticated linguistic features, such as readability metrics and stylometry metric should also be explored for this dataset. To fully understand the impact of each additional feature added it would be helpful to perform an ablation study, where every model would be trained and tested again with different feature combinations. It could also be possible to explore features impact thruguh a genetic based algorithm with different feature combinations for feature reduction as in study [55]. Since articles can be of the substantial lenght it can be very computationaly demanding to keep all of the relevant text when training the models, even large and impressive models such as BERT have a limit of 512 words per document. There are other large language models such as Longformer witch can process sequences of 4096 tokens, the use of such a large model could be benefitial with longer news articles [54]. However, in our work in an atempt to include the embeddings of each word, the embeddings were summed up into one embedding vector, which resulted in loss of information and results obtained by deep nerual networks are not as high as similar research. One solution to lenght news articles would be to include only parts of the article, as shown in research [45], where authors used only the first 300 words, arguing that the begining of the article will be the most emotion and information dense part. Another improvement could be made during the emotional analysis by usign multiple emotional lexicons instead of just one as shown in similar work [45]. Another issue could be that the used emotional lexicon is not domain speciffic, since the emotional values of articles were quite small. A reason for such low emotional values could be the lack of relevant words in the emotional lexicon. When dealing with tens of thousands of article ensuring the accuracy of the dataset through expert based verification presents a challenge. Manually fact-checking and correctly classifying each article as fake or real takes considerable time and effort, yet it would be highly benefitial and improve the quality of this and future research of the dataset. One of the directions of future work should be the deployment of the trained models in a publicy accesible format, such as a website or a mobile app. This would make the fake news detection tools more readly available. For example users could input a link to an article and the software would automaticaly preprocces the text and assess its thruthfullness, providing a practical application of research as shown in work [50]. Extending this idea ever futher would be the ability to evaluate text in different languages, making it accessible to more users. In our case it would be interesting to develop and avaluate a model for Croatian fake news detection. Another critique fake news datasets is their unrealistic representation as each article being fully accurate or fully inacurate witch has been pointed out by other researches as well [6]. There is a need to add muplitple levels of thruth to make the datasets more representative of the real word situations. Implementing a multi-level classification system would offer a more realistic representation of news articles accuracy, however it could introduce additional complexity compared to binary classification. Additionaly it would be valuable to explore the potential of the presented models as an ensammble method. There has been research on different ensamble and hybrid models for fake news detection, and the results achieved were promising [12].

6. Conclusion

This thesis explores the ability of machine learning and deep learning models to detect fake news in a climate change domain. Based on the research presented in Chapter 2. and Chapter 5., this study makes a contribution to the current research in the fake news detection in a climate change domain by testing and evaluating multiple classification methods. According to the results in Chapter 4. fake news holds more emotional value, but it is important to node that the average values of emotion in the dataset were relativley small. It was found that a fine-tuned BERT model performed the best, affirming it's value as a classification model for different domains. XGBoost model almost performed as well as BERT with a significantly quicker training time, showing that simpler ansamble methods can be almost as effective as transformers. Therefore, BERT and XGBoost models were shown to be effective in the fake detection task and they could potentially be improved with suggestions explained in Chapter 5. to develop a publicly available fake news detection tool. Climate change presents a global but polarizing issue. The development of tools for accurate and automated analysis of news articles is essential to help identify misinformation and uncover factual information.

References

[1] Nagi, K. (2018). *New Social Media and Impact of Fake News on Society* (SSRN Scholarly Paper 3258350). https://papers.ssrn.com/abstract=3258350

[2] Lahby, M., Aqil, S., Yafooz, W., & Abakarim, Y. (2022). Online Fake News Detection Using Machine Learning Techniques: A Systematic Mapping Study (pp. 3–37). https://doi.org/10.1007/978-3-030-90087-8_1

[3] Al-Asadi, M., & Tasdemir, S. (2022). *Using Artificial Intelligence Against the Phenomenon of Fake News: A Systematic Literature Review* (pp. 39–54). https://doi.org/10.1007/978-3-030-90087-8_2

[4] Mridha, M. F., Keya, A. J., Hamid, Md. A., Monowar, M. M., & Rahman, Md. S. (2021). A Comprehensive Review on Fake News Detection With Deep Learning. *IEEE Access*, *9*, 156151–156170. https://doi.org/10.1109/ACCESS.2021.3129329

[5] Allcott, H., & Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, *31*(2), 211–236. https://doi.org/10.1257/jep.31.2.211

[6] Oshikawa, R., Qian, J., & Wang, W. Y. (2020). *A Survey on Natural Language Processing for Fake News Detection* (arXiv:1811.00770). arXiv. https://doi.org/10.48550/arXiv.1811.00770

[7] Elyassami, S., Alseiari, S., ALZaabi, M., Hashem, A., & Aljahoori, N. (2022). Fake News Detection Using Ensemble Learning and Machine Learning Algorithms (pp. 149–162). https://doi.org/10.1007/978-3-030-90087-8_7

[8] Papakostas, D., Stavropoulos, G., & Katsaros, D. (2022). Evaluation of Machine Learning Methods for Fake News Detection. In M. Lahby, A.-S. K. Pathan, Y. Maleh, & W. M. S. Yafooz (Eds.), *Combating Fake News with Computational Intelligence Techniques* (pp. 163–183). Springer International Publishing. https://doi.org/10.1007/978-3-030-90087-8_8

[9] Przybyla, P. (2020). Capturing the Style of Fake News. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), Article 01. https://doi.org/10.1609/aaai.v34i01.5386

[10] Shu, K., Wang, S., & Liu, H. (2019). Beyond News Contents: The Role of Social Context for Fake News Detection. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 312–320. https://doi.org/10.1145/3289600.3290994

[11] Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019). dEFEND: Explainable Fake News Detection. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 395–405. https://doi.org/10.1145/3292500.3330935

[12] Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. International Journal of Information Management Data Insights, 1(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

[13] Mattern, J., Qiao, Y., Kerz, E., Wiechmann, D., & Strohmaier, M. (2021). FANG-COVID: A New Large-Scale Benchmark Dataset for Fake News Detection in German. In R. Aly, C. Christodoulopoulos, O. Cocarascu, Z. Guo, A. Mittal, M. Schlichtkrull, J. Thorne, & A. Vlachos (Eds.), Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER) (pp. 78–91). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.fever-1.9

[14] Snijders, C., Conijn, R., Fouw, E. de, & Berlo, K. van. (2023). Humans and Algorithms Detecting Fake News: Effects of Individual and Contextual Confidence on Trust in Algorithmic Advice. International Journal of Human–Computer Interaction. https://www.tandfonline.com/doi/full/10.1080/10447318.2022.2097601

[15] Himdi, H., Weir, G., Assiri, F., & Al-Barhamtoshy, H. (2022). Arabic Fake News Detection Based on Textual Analysis. *Arabian Journal for Science and Engineering*, *47*(8), 10453–10469. https://doi.org/10.1007/s13369-021-06449-y

[16] Plotkina, D., Munzel, A., & Pallud, J. (2020). Illusions of truth—Experimental insights into human and algorithmic detections of fake online reviews. *Journal of Business Research*, *109*, 511–523. https://doi.org/10.1016/j.jbusres.2018.12.009

[17] Kuzmin, G., Larionov, D., Pisarevskaya, D., & Smirnov, I. (2020). Fake news detection for the Russian language. In A. Aker & A. Zubiaga (Eds.), Proceedings of the 3rd International Workshop on Rumours and Deception in Social Media (RDSM) (pp. 45–57). Association for Computational Linguistics. https://aclanthology.org/2020.rdsm-1.5

[18] Rubin, V. L., & Conroy, N. (2012). Discerning truth from deception: Human judgments and automation efforts. *First Monday*. https://doi.org/10.5210/fm.v17i3.3933

[19] Schuster, T., Schuster, R., Shah, D. J., & Barzilay, R. (2020). The Limitations of Stylometry for Detecting Machine-Generated Fake News. Computational Linguistics, 46(2), 499–510. https://doi.org/10.1162/coli_a_00380

[20] Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). *Automatic Detection of Fake News* (arXiv:1708.07104). arXiv. https://doi.org/10.48550/arXiv.1708.07104

[21] Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. https://doi.org/10.1162/tacl_a_00349

[22] Kaliyar, R. K., Goswami, A., & Narang, P. (2021). FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimedia Tools and Applications*, *80*(8), 11765–11788. https://doi.org/10.1007/s11042-020-10183-2

[23] Meddeb, P., Ruseti, S., Dascalu, M., Terian, S.-M., & Travadel, S. (2022). Counteracting French Fake News on Climate Change Using Language Models. *Sustainability*, *14*, 11724. https://doi.org/10.3390/su141811724

[24] Singhal, S., Shah, R. R., Chakraborty, T., Kumaraguru, P., & Satoh, S. (2019). SpotFake: A Multi-modal Framework for Fake News Detection. *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, 39–47. https://doi.org/10.1109/BigMM.2019.00-44

[25] Chu, S. K. W., Xie, R., & Wang, Y. (2021). Cross-Language Fake News Detection. *Data and Information Management*, *5*(1), 100–109. https://doi.org/10.2478/dim-2020-0025

[26] Dou, Y., Shu, K., Xia, C., Yu, P. S., & Sun, L. (2021). User Preference-aware Fake News Detection. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2051–2055. https://doi.org/10.1145/3404835.3462990

[27] Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., & Gao, J. (2018). EANN: 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018. KDD 2018 - Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 849–857. https://doi.org/10.1145/3219819.3219903

[28] Patel, M., Padiya, J., & Singh, M. (2022). Fake News Detection Using Machine Learning and Natural Language Processing (pp. 127–148). https://doi.org/10.1007/978-3-030-90087-8_6

[29] NASA Faked the Moon Landing—Therefore, (Climate) Science Is a Hoax: An Anatomy of the Motivated Rejection of Science—Stephan Lewandowsky, Klaus Oberauer, Gilles E. Gignac, 2013. (n.d.). Retrieved 6 August 2024, from https://journals.sagepub.com/doi/abs/10.1177/0956797612457686

[30] Hong, S. C. (2020). Presumed Effects of “Fake News” on the Global Warming Discussion in a Cross-Cultural Context. *Sustainability*, *12*(5), Article 5. https://doi.org/10.3390/su12052123

[31] Gruener, S. (2024). Determinants of Gullibility to Misinformation: A Study of Climate Change, COVID-19 and Artificial Intelligence. *Journal of Interdisciplinary Economics*, *36*(1), 58–78.

[32] Bramoullé, Y., & Orset, C. O. (2018). Manufacturing Doubt. *Journal of Environmental Economics and Management*, *90*, 119–133. https://doi.org/10.1016/j.jeem.2018.04.010

[33] Kapantai, E., Christopoulou, A., Berberidis, C., & Peristeras, V. (2020). A systematic literature review on disinformation: Toward a unified taxonomical framework. *New Media & Society*, *23*, 146144482095929. https://doi.org/10.1177/1461444820959296

[34] Zhou, Y., & Shen, L. (2022). Confirmation Bias and the Persistence of Misinformation on Climate Change. *Communication Research*, *49*(4), 500–523. https://doi.org/10.1177/00936502211028049

[35] Drummond, C., Siegrist, M., & Árvai, J. (2020). Limited effects of exposure to fake news about climate change. *Environmental Research Communications*, *2*(8), 081003. https://doi.org/10.1088/2515-7620/abae77

[36] Omar Bali, A. (2023). Raising Climate Change Awareness Across Twitter. *The Journal of Environment & Development*, *32*(4), 370–391. https://doi.org/10.1177/10704965231205020

[37] Cheng, H., & Gonzalez-Ramirez, J. (2021). Trust and the Media: Perceptions of Climate Change News Sources Among US College Students. *Postdigital Science and Education*, *3*(3), 910–933. https://doi.org/10.1007/s42438-020-00163-y

[38] Al-Rawi, A., O'Keefe, D., Kane, O., & Bizimana, A.-J. (2021). Twitter's Fake News Discourses Around Climate Change and Global Warming. *Frontiers in Communication*, *6*. https://doi.org/10.3389/fcomm.2021.729818

[39] Dahal, B., Kumar, S. A. P., & Li, Z. (2019). Topic modeling and sentiment analysis of global climate change tweets. *Social Network Analysis and Mining*, *9*(1), 24. https://doi.org/10.1007/s13278-019-0568-8

[40] Gautam, A., V, V., & Masud, S. (2021). *Fake News Detection System using XLNet model with Topic Distributions: CONSTRAINT@AAAI2021 Shared Task* (arXiv:2101.11425). arXiv. https://doi.org/10.48550/arXiv.2101.11425

[41] Upadhyaya, A., Fisichella, M., & Nejdl, W. (2023). A Multi-Task Model for Sentiment Aided Stance Detection of Climate Change Tweets. *Proceedings of the International AAAI Conference on Web and Social Media*, *17*, 854–865. https://doi.org/10.1609/icwsm.v17i1.22194

[42] Chu, J., Zhu, Y., & Ji, J. (2023). Characterizing the semantic features of climate change misinformation on Chinese social media. *Public Understanding of Science*, *32*(7), 845–859. https://doi.org/10.1177/09636625231166542

[43] Falkenberg, M., Galeazzi, A., Torricelli, M., Di Marco, N., Larosa, F., Sas, M., Mekacher, A., Pearce, W., Zollo, F., Quattrociocchi, W., & Baronchelli, A. (2022). Growing polarization around climate change on social media. *Nature Climate Change*, *12*(12), 1114–1121. https://doi.org/10.1038/s41558-022-01527-x

[44] Upadhyaya, A., Fisichella, M., & Nejdl, W. (2023). Intensity-Valued Emotions Help Stance Detection of Climate Change Twitter Data. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 6246–6254. https://doi.org/10.24963/ijcai.2023/693

[45] Ghanem, B., Rosso, P., & Rangel, F. (2020). An Emotional Analysis of False Information in Social Media and News Articles. *ACM Trans. Internet Technol.*, *20*(2), 19:1-19:18. https://doi.org/10.1145/3381750

[46] Zhou, L., Tao, J., & Zhang, D. (2023). Does Fake News in Different Languages Tell the Same Story? An Analysis of Multi-level Thematic and Emotional Characteristics of News about COVID-19. *Information Systems Frontiers*, *25*(2), 493–512. https://doi.org/10.1007/s10796-022-10329-7

[47] Momeni Rouchi, P. (2023). *Fake news detection performance analysis by incorporation of sentiment analysis* [Thesis, Technische Universität Wien]. https://doi.org/10.34726/hss.2023.105745

[48] Poddar, K., Amali D., G. B., & Umadevi, K. S. (2019). Comparison of Various Machine Learning Models for Accurate Detection of Fake News. *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*, *1*, 1–5. https://doi.org/10.1109/i-PACT44901.2019.8960044

[49] Patwa, P., Sharma, S., Pykl, S., Guptha, V., Kumari, G., Akhtar, M. S., Ekbal, A., Das, A., & Chakraborty, T. (2021). *Fighting an Infodemic: COVID-19 Fake News Dataset* (Vol. 1402, pp. 21–29). https://doi.org/10.1007/978-3-030-73696-5_3

[50] Anil, Mr. (2023). Evaluation of Deep Learning Models for Fake News Detection. *International Journal of Computational Intelligence Research (IJCIR)*, *19*(1), 1–11. https://doi.org/10.37622/IJCIR/19.1.2023.1-11

[51] Wu, J., & Ye, X. (2023). FakeSwarm: Improving Fake News Detection with Swarming Characteristics. *Natural Language Processing and Machine Learning*, 175–187. https://doi.org/10.5121/csit.2023.130815

[52] Santhiya, P., Kavitha, S., Aravindh, T., Archana, S., & Praveen, A. V. (2023). Fake News Detection Using Machine Learning. *2023 International Conference on Computer Communication and Informatics (ICCCI)*, 1–8. https://doi.org/10.1109/ICCCI56745.2023.10128339

[53] Balakrishnan, V., Zing, H. L., & Laporte, E. (2023). COVID-19 INFODEMIC – UNDERSTANDING CONTENT FEATURES IN DETECTING FAKE NEWS USING A MACHINE LEARNING APPROACH. *Malaysian Journal of Computer Science*, *36*(1), Article 1. https://doi.org/10.22452/mjcs.vol36no1.1

[54] Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The Long-Document Transformer* (arXiv:2004.05150). arXiv. https://doi.org/10.48550/arXiv.2004.05150

[55] Ogundokun, R. O., Arowolo, M. O., Misra, S., & Oladipo, I. D. (2022). Early Detection of Fake News from Social Media Networks Using Computational Intelligence Approaches. In M. Lahby, A.-S. K. Pathan, Y. Maleh, & W. M. S. Yafooz (Eds.), *Combating Fake News with Computational Intelligence Techniques* (pp. 71–89). Springer International Publishing. https://doi.org/10.1007/978-3-030-90087-8_4

[56] Dubey, Y., Wankhede, P., Borkar, A., Borkar, T., & Palsodkar, P. (2022). Framework for Fake News Classification Using Vectorization and Machine Learning. In M. Lahby, A.-S. K. Pathan, Y. Maleh, & W. M. S. Yafooz (Eds.), *Combating Fake News with Computational Intelligence Techniques* (pp. 327–343). Springer International Publishing. https://doi.org/10.1007/978-3-030-90087-8_16

[57] 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). Springer International Publishing. https://doi.org/10.1007/978-3-319-69155-8_9

[58] Zhou, X., Jain, A., Phoha, V. V., & Zafarani, R. (2020). Fake News Early Detection: A Theory-driven Model. *Digital Threats*, *1*(2), 12:1-12:25. https://doi.org/10.1145/3377478

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

[60] Reis, J. C. S., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Supervised Learning for Fake News Detection. *IEEE Intelligent Systems*, *34*(2), 76–81. IEEE Intelligent Systems. https://doi.org/10.1109/MIS.2019.2899143

[61] Cui, L., Wang, S., & Lee, D. (2020). SAME: Sentiment-aware multi-modal embedding for detecting fake news. *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 41–48. https://doi.org/10.1145/3341161.3342894

[62] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. http://arxiv.org/abs/1810.04805

[63] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1532–1543. https://doi.org/10.3115/v1/D14-1162

Sažetak

Otkrivanje lažnih vijesti u diskursu o ekologiji i klimatskim promjenama

Ovaj diplomski rad istražuje primjenu različitih modela strojnog učenja i dubokog učenja za otkrivanje lažnih vijesti u domeni klimatskih promjena. Implementirano je i testirano više modela s različitim tekstualnim reprezentacijama i dodatnim značajkama kao što su sentimentalne, emocionalne i sintaktičke karakteristike. Rezultati pokazuju da lažne vijesti uglavnom imaju veći emocionalni sadržaj, iako su prosječne emocionalne vrijednosti u skupu podataka bile relativno niske. Rad je otkrilo da fino podešeni BERT model nadmašuje ostale, pokazujući se kao učinkovit klasifikator u različitim domenama. Model XGBoost također je dao dobre rezultate, gotovo jednake BERT-u. Razvijanje alata za točnu i automatsku analizu novinskih članaka u domeni klimatskih promjena ključno je za prepoznavanje dezinformacija.

Lažne vijesti, obrada prirodnog jezika, strojno učenje, duboko učenje, klimatske promjene, emotivna analiza.

Summary

Detecting fake news in the discourse on ecology and climate change

This thesis explores the application of different machine learning and deep learning models to detect fake news in the domain of climate change. Multiple models with different textual representations and additional sentimental, emotional and syntactic features have been implemented and tested. The results show that fake news generally have a higher emotional value, although the average emotional values in the dataset were relatively low. This thesis found that a fine-tuned BERT model outperforms the other models, proving to be an effective classifier in different domains. The XGBoost model also gave good results, almost equal to BERT. Developing tools to accurately and automatically analyze climate change news articles is critical to identifying misinformation.

Fake news, natural language processing, machine learning, deep learning, climate change, emotional analysis.