 USING MACHINE LEARNING IN DETECTING FAKE NEWS

ȘTEFAN BOLOTĂ¹, MIRCEA ASANDULUI ii

Abstract: In a world that has been greatly affected by the Coronavirus pandemic and more recently by the armed conflict between Russia and Ukraine, the flow of information is constantly increasing and at the same time the veracity of this information raises a big concern, and this makes the topic of fake news a problem of major interest. Our paper proposes a tool for fake news detection using different models of machine learning developed over a Fake News Corpus. Neural networks have proven to be the most effective method, reaching an accuracy of over 90%, but also Naive Bayes can be an excellent solution for classifying text data. Besides these two, we also developed and analyzed other models based on Naive Bayes and k-Nearest Neighbors. The results are promising and show that the problem of fake news can be managed by machine learning algorithms.

Keywords: fake news detection, neural networks, machine learning, artificial intelligence, natural language processing, Naive Bayes.

JEL Classification: C45, C63

1. INTRODUCTION

With so much of our lives spent online on social media, more and more people tend to seek and consume news and other information from social media, rather than traditional media sources as a more accessible route.

In a world that has been hit hard by the Coronavirus pandemic, and where many countries have restricted the right to free movement, forcing people to stay indoors, the consumption of information has increased considerably from all media sources. At the same time, with the entire world watching the armed conflict between Russia and Ukraine, the flow of news is increasing.

¹ Alexandru Ioan Cuza University of Iași, Faculty of Economics and Business Administration, Iași, Romania, stefan_bolota17@yahoo.com
ii Alexandru Ioan Cuza University of Iași, Faculty of Economics and Business Administration, Iași, Romania, mircea.asandului@uaic.ro
But the authenticity of the information has become a major issue affecting the whole society. On social media, information is circulating so fast and so intensely. Due to this it has a huge capacity to make an impact in the real world, affecting millions of people in a matter of minutes.

Although this is a major problem, still there is no concrete solution for the problem. In the news industry, fake news has become the main topic of discussion, as most content is questioned for its veracity due to the constant, alarming spread of fake news. This is even a more urgent matter especially during the pandemic period, when people are scared and tend to believe any conspiracy or any information that has no substantiated basis. Also, during this pandemic period, a plethora of myths have appeared in the media, from different ways to get rid of the Coronavirus to the darkest conspiracies, from Bill Gates stealing humanity's data, 5G antennas causing the disease to vaccine-implanted chips.

Society’s problems grow with man's inability to distinguish accurate, authentic news from fake news. According to Anderson (2017) young people are generally known to be "all-knowing" compared to their parents, but when it comes to determining whether a news story is fake or not, they prove to be just as confused as the rest of society. One of the biggest problems facing the field of media, Journalism, especially the digital area, is the out of control spread of fake news. The fake news, which is the subject of the second problem, is amplified and 'maintained' by big companies like Google, Apple, Facebook, and Amazon who decide who publishes, what content is published and how this fake news is monetized.

Fake news has become so widespread and widely circulated that journalists publish their own content on their personal social media accounts, thus manipulating society and spreading the content with exceptional ease.

With all this in mind, we have concluded that a study is needed, and above all a way to bring down this "conspiracy" of fake news, the first step being to identify it. With the exacerbated evolution of technology, it is necessary to sound the alarm on our response to the effects that technology produces. Society deserves to live by honest rules, it deserves to be informed correctly and on time.

Establishing the veracity of information online is therefore a current challenge, requiring attention, regulation and active monitoring of digital content disseminated by media entities involved in supporting the way information is presented and shared between people, on the internet, including search engines and social networking platforms.
Our goal is, using Data Mining and Machine Learning techniques, to develop a tool that can verify the veracity of the information in a manner as simple as possible and in the same time very accurate.

To get to the core of the issue and to better understand the impact of fake news on humanity, we will present concrete reports and facts that reveal the negative effects of this phenomenon, the characteristics of fake news and how to identify it. It should be noted that fake news is not just about spreading totally false stories. This phenomenon has several "manifestations": it is possible for a news item to be false simply by omitting certain issues, by focusing on a single aspect and forgetting the context, by reformulating certain lines given by certain people, by changing certain facts, etc.

Fake news is often used for financial or political purposes and is associated with propaganda aimed at spreading misleading information to promote a political interest or point of view. We consider that the main characteristics of this phenomenon include the fact that the news is inaccurate, the content is optimized for distribution or sharing, and that the information is designed to mask or distort emotions by emphasizing bias or discrimination.

If we look at it from a journalistic point of view, the specific elements are structural elements such as the headline, the body of the news, the images, etc. On the side of intent to deceive we find the desire to provoke the reader from a political/ideological or financial point of view.

Further on, in the following section, we will present a literature review, analyzing some of the existing studies and their relation to the problem of identifying fake news. After this, we will discuss the methods and mechanisms used to develop machine learning models, as well as the data used to train the models. In the last part of this paper, we will present the results obtained and some conclusions that emerge from our analysis.

2. LITERATURE REVIEW

To identify fake news, it is not only necessary to think about whether it might happen, to check several sources, to check the date the news is posted or to check the author.

Based on the features, different models for recognizing fake news exist. The first model is the knowledge-based model. This model is done either by journalists or experts in the field or by majority vote. Majority voting refers to the idea that if a majority claims that a certain content is false, then it must be so. As an example, we
can consider the case of Coronavirus, focusing on the analysis of the social network Facebook. Majority voting would have a greater impact in Facebook groups, as people socialize and exchange different opinions. In the context of participating in a discussion outside my area of knowledge, not having sufficient information on that topic, I tend to follow the herd effect and conform to the majority.

The second pattern is called the automatic verification pattern. This model makes use of a huge database that contains information about any field, thing, etc.

The third model is the style-based model: fake news generally uses a specific style that plays on the reader's emotions to change their behavior. Another commonly used style is sensationalism which arouses curiosity in the reader that makes him click on the news. In general, these sensational news stories come from the world of social media. Journalists use words such as 'sensational', 'exclusive', 'you won't believe it...' etc. These formulas arouse curiosity mainly through their wording and not necessarily through the subject of the news. In other words, someone may access a news story, only persuaded by the wording of the headline, and not by the subject matter.

The most effective and appropriate model for recognizing fake news is based on content. The knowledge-based model is from a practical point of view very difficult to implement, being expensive and needing a huge database that has to be manipulated.

Reis et al. (2019) use machine learning techniques for newsfeed articles related to the 2017 US election. The algorithms evaluated were k-nearest neighbors (kNN), Random Forest, Support Vector Machines (SVM) and Extreme Gradient Boosting (XGBoost). To develop these algorithms a lot of hand-crafted features were used, such as language features bag-of-words, POS tagging and others for a total of thirty-one different features), lexical feature( number of unique words and their frequencies), psychological features (built using Linguistic Inquiry and Word Count which is a specific dictionary built by a specialized text extraction program) and semantic features (toxic score obtained through Google API). Many other features, extracted from source and social metadata, were also used.

Their results are shown in Figure 1. They also show that XGBoost is good at selecting texts that need to be manually verified, meaning that texts classified as trustworthy are indeed trustworthy and thus reducing the number of texts that need to be manually verified. This model is limited by the fact that they use metadata that is not always available.
Perez-Rosas et al. (2018) used almost the same set of features but also used a linear SVM-based model and worked on a different dataset. The models that had the best performances in their case were XGBoost and Random Forrest.

Another interesting study uses a hybrid model (CSI) to detect fake news. Ruchansky et al. (2017) used a hybrid network, combining features obtained from news content and metadata, such as social engagement, into a single network.

To do this, they used a recurrent neural network (RNN) for extracting news timing features and a fully connected network for social features. For features extracted from text they used the dov2vec library. The results of the two networks are then concatenated and used for the final classification.

Yang et al. (2018) used a convolutional neural network (CNN) that uses images found in articles to do classification. They used a Kaggle dataset containing fake news, in addition they scraped real news from trusted sources such as the New York Times and Washington Post.

Their network consists of two branches: a text branch and an image branch (Figure 2). The text branch is then divided into two sub-branches: text explicit: information derived from the text, such as the length of the news, and the text latent sub-branch, which is basically the content of that news, limited to 1000 words.

**Figure 1 CSI model**
The branch dealing with image processing also consists of two sub-branches, one containing information such as image resolution or the number of people present in the image, and the second sub-branch using a convolutional neural network on the image itself. The results support the hypothesis that using images generates better results.

Gilda (2017) emphasizes the importance of Natural Language Processing (NLP) in identifying false or incorrect information. Using Term Frequency – Inverse Document Frequency (TF-IDF) of bigrams and probabilistic detection of context-free grammar. They used Bi-Gram Count Vectorizer (TF-IDF) and Probabilistic Context-Free Grammar (PCFG), the study detects false information. At the same time, the dataset from more than one class of algorithms was examined to find a better model. The bigram count vector was loaded directly into a stochastic top-down model that identifies items that are not credible with 71.2% accuracy.

Shu et al. (2017) focus on fake news on social networks presenting a data mining perspective that includes the characterization of fake news in psychology and social theories. Their paper analyzes two main factors responsible for the widespread user acceptance of fake messages, which is naive realism and confirmation bias. A general two-phase data mining framework is proposed which includes feature extraction and modeling, dataset analysis and confusion matrix for false news detection.

Parikh and Atrey (2018) point out that social networking sites process news mainly in three ways. The first option is by multilingual text; this is analyzed using computational linguistics, which focuses semantically and systematically on how the text was made. Since most publications are in text form, their analysis requires enough work to draw some relevant conclusions. Another manner is multimedia,
when several forms of media are integrated into a single publication. This can include audio, video, images, and graphics and is incredibly attractive and attracts the viewer's attention without having to consider much text. The last mentioned way is by hyperlinks that allow the author to make posts referring to various sources and thus gain the trust of viewers. In practice, references are made to other social media sites and screenshots are inserted.

3. DATA AND METHODOLOGY

In this chapter we will present information about the used dataset and the methodology used to develop an application that is able to predict fake news. The dataset we use is called the Fake News Corpus and is an open-source dataset composed of millions of news articles that have been extracted from a list of 1001 domains. This corpus is intended for the development of data mining algorithms for the purpose of fake news recognition.

Being still under development, the public version includes over 9 million articles (745 out of 1001 domains). Out of the nine million data we used for the training part only 200 thousand news that are associated with de labels reliable, fake, unreliable, clickbait, conspiracy.

Data mining models for text classification

We take into consideration three classification methods to classify news, namely neural networks, the Naïve Bayes classifier and the k-NN classifier; further we present briefly how these methods work and the principles behind them.

Neural networks are a branch of artificial intelligence that simulates brain activity within a program to create a machine learning model for the computer. In practice, neural networks are used for classification and regression problems.

Within a neural network the term perceptron appears (Fig. 3). A perceptron is an artificial neuron model in which the input signals are summed, and the output signal appears only if the sum exceeds the threshold $\theta$.

![Figure 3 Structure of a perceptron](image-url)
The purpose of the perceptron is to classify an instance into one of two classes (0 or 1). The perceptron has \( n \) inputs \((x_1, x_2, \ldots, x_n)\) and one output \( y \), and \( w_i \) is the weight for each input \( x_i \).

Using the standard (single layer) perceptron is not always a happy one, as it is unable to implement some functions, such as the logical XOR function or larger networks, thus creating an output that very sharply separates the input space into two halves (Fig. 4).

This is why it was necessary to introduce multilayer perceptron in which the output function is no longer binary, but a real value between 0 and 1 that can be interpreted as a probability. (Fig. 5) The output here is smooth, continuous, thus solving the problem of very abrupt switching from one value to another.

The multilayer perceptron is a feed-forward neural network with one or more hidden layers, as it can been observed below.

This example of a neural network is a fully connected network, i.e. there is an arc from every perceptron on layer \( i \) to all perceptrons on layer \( i+1 \).

Another important parameter in neural networks is the activation function which, for the multilayer perceptron, must be non-linear.

The way the neural network creates the learning model is based on an algorithm to adjust the network weights to reduce the difference between the actual
output \((y)\) and the desired output \((y_d)\). One such algorithm is the back-propagation algorithm that has two stages.

First, the network receives the input data as a vector \((x_1, x_2, \ldots, x_n)\) and sends the signal forward, step by step in each layer, until the last layer generating the output is reached.

The second step is to propagate the error from the output layer to the input layer by adjusting the network weights:

\[
\Delta w_{jk} = \alpha \cdot y_j \cdot [y_k (1 - y_k)] \cdot e_k \\
\Delta w_{ij} = \alpha \cdot x_i \cdot [y_j (1 - y_j)] \cdot \sum_k \delta_k w_{jk}
\]

where \(\Delta w\) – weight correction, \(\alpha\) – learning rate, \(x\) – input, \(e\) – error \((y_d - y)\)

The Naive Bayes algorithm is a classification method based on Bayes' theorem. This algorithm assigns a particular label to new instances based on the probabilities calculated for each label. Specifically, if we have a model that needs to classify a new instance as True or False, the algorithm will calculate the probability that the instance is True and the probability that the instance is False, and then compare the probabilities and choose the higher one.

Bayes' theorem calculates the probability of an event occurring, considering the probability of another event that has already occurred and has the following form:

\[
P( A | B) = \frac{P( B | A) \cdot P( A)}{P(B)}
\]

To calculate these probabilities, we need a data set that is divided into two parts, a set of features \((D)\), and an output \(y\) (classification label). Thus, in our case, the formula becomes:

\[
P( y | D) = \frac{P( D | y) \cdot P( y)}{P( D)}
\]

The naivety of the Naive Bayes algorithm comes from the naive assumption of data independence, i.e., each pair of classified features is independent of each other. In this case, we can have the following formula:

\[
P( A, B) = P( A) \cdot P( B)
\]

Using last formula, we can write Bayes' formula in the following form:

\[
P( y | D) = P( y | x_1, \ldots, x_n) = \frac{P( x_1 | y) P( x_2 | y) \ldots P( x_n | y)}{P( x_1) P( x_2) \ldots P( x_n)}
\]
To create a classification model of the data, we calculate the posterior probabilities for all possible values of the variable class $y$ and choose the output with the maximum probability. We can write this in the following form:

$$P(y|x_1, \ldots, x_n) = \frac{P(y) \cdot \prod_{i=1}^{n} P(x_i|y)}{P(x_1) \cdot P(x_2) \cdot \cdots \cdot P(x_n)}$$

Finally, we have to calculate the class probability $P(Y=y)$ and the conditional probabilities $P(x_i|y)$. In total, we have calculate $2^n+1$ probabilities, where $n$ is the number of features.

**K-Nearest Neighbors (K-NN)** is a classification and regression algorithm that ranks instances based on their k nearest neighbors. This classifier starts from the idea that similar things are close to each other, and in this sense, similarity would be calculated as the distance between the two points. The smaller the distance, the more similar things are.

There are several ways we can calculate the distance between two points, but it depends on the problem we want to solve. However, in practice, Euclidean distance is most often used.

If $p=(p_1, p_2, \ldots, p_n), q=(q_1, q_2, \ldots, q_n)$ points in an $n$-dimensional Euclidean space, then the Euclidean distance is calculated according to the formula:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

The success of the algorithm lies in the right choice of the parameter $k$. This choice can be made by running the algorithm several times with different values for $k$ and selecting the parameter that reduces the error, while maintaining the algorithm’s ability to make accurate predictions. This method is also called the elbow method. This algorithm is a simple and easy algorithm to implement and does not require model building or additional assumptions. However, the k-NN algorithm can become significantly slower as the dataset grows.

### 4. Results and Discussions

We randomly extracted a sample of 200 thousand articles, and we used this part of the data set to train the algorithms. In order to do this, we created a pipeline to put together the whole data processing process, starting from the news given as
input in the original format, then processing it, then transforming it into a different representation, so that finally this representation can train the model.

Another important aspect to be mentioned is the length of time over which the training took place. For this dataset, which also used the pipeline described above, only the pre-processing and vectorization of the text took about 12 hours (the length of the news stories was long enough), and the actual training lasted per classifier.

To train the models, we used 80% of our dataset, and the remaining 20% we used for the testing process to obtain the score for the three classifiers.

In terms of training time, the model created using Naive-Bayes is the fastest, followed by k-NN, and finally neural networks. However, this ranking changes when we talk about model accuracy, the results are shown in Table 1.

### Table 1 Classification results on the training dataset

<table>
<thead>
<tr>
<th></th>
<th>Neural Networks</th>
<th>Naïve-Bayes</th>
<th>k-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1-score</td>
</tr>
<tr>
<td>Fake</td>
<td>0.86</td>
<td>0.98</td>
<td>0.92</td>
</tr>
<tr>
<td>Reliable</td>
<td>0.98</td>
<td>0.87</td>
<td>0.92</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.92</td>
<td>0.93</td>
<td>0.92</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.93</td>
<td>0.92</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Next, we evaluated our trained models and for this we extracted a new dataset consisting of about 750 thousand instances. On this dataset, we assessed each of the three models and obtained similar scores (Table 2), which shows that the models we created are quite good.

### Table 2 Classification results on the test dataset

<table>
<thead>
<tr>
<th></th>
<th>Neural Networks</th>
<th>Naïve-Bayes</th>
<th>k-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1-score</td>
</tr>
<tr>
<td>Fake</td>
<td>0.94</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>Reliable</td>
<td>0.92</td>
<td>0.94</td>
<td>0.93</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.93</td>
<td>0.93</td>
<td>0.93</td>
</tr>
</tbody>
</table>

To conclude, training was one of the most difficult steps in the modeling part of a classifier, especially in terms of the execution time of the different tests we did, but also in terms of the limited hardware components. As it can be observed, the Neural Networks have the best classification results, classifying correctly more than 9 out of 10 news. For the last two classifiers, we did a graphical analysis measuring the scalability and performance of the models (Figure 7, 8 and 9).
Moreover, in practice it turns out that Bayes is much faster than 3-NN when it comes to classifying a new instance, since in the case of the latter all distances between the new instance and the data with which the model was trained have to be computed.

Figure 7 Training and cross-validation score analysis for Naive-Bayes and 3-NN

Figure 8 Scalability analysis of the two models

Figure 9 Performance analysis of the two models
As it can be observed from the graphics above, the Naive-Bayes classifier performs better than 3-NN on our dataset, even though the time required for the 3-NN model to train with different training dataset sizes is smaller than that of the Naïve-Bayes classifier.

5. CONCLUSIONS

The aim of this work was to develop a desktop application capable of detecting fake news. For the development of the application, we used three classifiers in order to see which machine learning model fits better the data. In building the classifiers we started from a dataset of about nine million records. We extracted two data sets, the first was used for training the algorithm and the second one was used to evaluate the algorithms’ performance. On average we achieved an accuracy of 90%, which tells us that our models can correctly classify nine out of 10 news stories.

In our future work, to improve the created solution, we will create a database to keep track of all the new articles that have been entered for classification, and in the case of a story that has been classified with a probability of less than 75%, the user could have the option to choose what kind of story it is. In this way, when the classifier gathers a certain number of new items in that database, it will train again with the new data, thus achieving dynamic learning.
REFERENCES