

# Application of Neural Networks to Identify of Fake News

Iryna Afanasieva, Nataliia Golian, Vira Golian, Artem Khovrat and Kostiantyn Onyshchenko

*Kharkiv National University of Radio Electronics, 14, Nauky, Ave., Kharkiv, 61166, Ukraine*

## Abstract

The problem of determining information reliability is especially acute in connection with significant social unrest, which often manifests itself in recent times. At the same time, news resources use aggregated data from social networks, which is increasingly difficult to verify. Various classification algorithms make it possible to speed up this process, however, in critical conditions, the accuracy of the forecast becomes low. For the current study, it was determined to investigate the effectiveness of using neural networks to detect fake news. To increase the accuracy of the classification, it was decided to create a data preprocessing algorithm based on the basic principles of natural language processing. Based on the results of this study, linguistic patterns of fake news were identified. The resulting templates became the basis for data preprocessing. The features of convolutional and recurrent neural networks and their modifications for the analysis of text data are disclosed. To compare certain models, a set of indicators characterizing the efficiency of the algorithms was chosen. The classification accuracy of these models was tested on data related to the election of the President of the United States and the large-scale invasion of the Russian Federation into the territory of Ukraine. The resulting indicator of the effectiveness of the classification of fake news allows us to state the feasibility of using certain modifications of both models to verify the reliability of the information.

## Keywords

Classification, CNN, information reliability, LSTM, Neural Network, RNN

## 1. Introduction

According to the Cambridge Dictionary: "fake news" – false stories that appear to be news, spread on the internet or using other media, usually created to influence political views or as a joke [1]. Their history is quite long, however, with the growing popularity of social networks, especially anonymous ones, this problem has become acute in front of world society. Its catalyst is also the development of technologies aimed at adjusting video and audio information, and technologies for creating bots.

Before the beginning of the full-scale invasion of the Russian Federation on the territory of Ukraine, a vivid example of manipulation with fakes was the information about the spread of the coronavirus from China, in particular the story about eating bats and the global conspiracy regarding the vaccine [2]. Before this story, there were reports of news manipulation during the election campaign in India, where according to some sources, 40% of the information was fake [3]; and during the presidential elections in Brazil in 2019, where 11,957 viral messages were distributed using the WhatsApp network, of which 42% contained information that was deemed false by regulatory authorities [4].

As of today, given the general political destabilization in the world, the number of fakes has increased both in absolute and relative terms. Some of them can be attributed to the incorrect subjective perception of real information, but a significant number is the result of military

---

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine  
EMAIL: iryna.afanasieva@nure.ua (I. Afanasieva); nataliia.golian@nure.ua (N. Golian); vira.golan@nure.ua (V. Golian);  
artem.khovrat@gmail.com (A. Khovrat); kostiantyn.onyshchenko@nure.ua (K. Onyshchenko)  
ORCID: 0000-0003-4061-0332 (I. Afanasieva); 0000-0002-1390-3116 (N. Golian); 0000-0001-5981-4760 (V. Golian) 0000-0002-1753-8929 (A. Khovrat); 0000-0002-7746-4570 (K. Onyshchenko)



© 2023 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

propaganda. If in the case of elections, the reason for the spread of fake news was to influence opinions, now it is aimed at demoralizing the population and the military to suppress their opposition. In addition, there are attempts to influence people who are not parties to a full-scale war to discredit the aid provided. The largest channels of such information are Twitter and Telegram [5]. In some cases, hacker intrusions on the sites of information sources and further data substitution are possible. However, this case is outside the scope of current work and is relevant to data transfer models [6]

In general, many initiatives have been created to combat such news, even before the current events, for example, there is the French law against the manipulation of information, which was adopted to combat the discrediting of immigrants and the European Union after the announcement of the Brexit results [7]. This law states that Platforms that exceed a certain number of visits per day must have a legal representative in France and publish their algorithms, while any sponsored content must be reported by publishing the author's name and the amount paid. The law also requires judges to qualify fake news based on the following three criteria:

- transparency;
- deliberate distribution on a mass scale;
- leading to violations of public order or compromising election results.

At the same time, the initial decision is made by a specially created ethics committee.

The creation of a legal framework for the regulation of fake information is also observed in Ukraine, for instance, there is Article 259 of the Criminal Code, which regulates responsibility for knowingly false information about a threat to the safety of citizens.

The global trend towards combating such information is generally positive, but the problem lies in determining whether the information is fake or not. If the decision is based not on facts, but on expert assessment, then such a situation can be considered manipulative. This situation is observed in the Russian Federation, Iran, and the DPRK.

For democratic countries, the process of identifying fake news can be automated using artificial intelligence. Initially, naïve Bayes or SVM was used as the basic method, but with the development of neural networks, convolutional and recurrent neural networks gained the most popularity.

It is worth noting that in view of numerous studies, these models can give different results depending on the subject area [8, 9]. Therefore, it was decided to consider the effectiveness of using these models to identify fake news, to optimize their selection when building an appropriate software system.

## 2. Domain analysis

Firstly, the task of identifying fake news concerns the process of natural language processing (NLP), therefore, before formalizing the purpose of this work, we will consider several basic concepts that will be used in the future.

Let's start with TF-IDF characteristics. Software systems are not able to directly process text information, therefore, each text should acquire a quantitative form. There are several basic techniques for this type of conversion, but currently, TF-IDF is the basic one. TF – short for term frequency – is the frequency of each used word (more precisely, a set of words that are more appropriate to call terms). IDF – short for inverse document frequency – is the inverse number of terms per document. In general, the TF-IDF indicator indicates how rare a certain term is. For example, interjections, conjunctions, or exclamations will be the most common and, accordingly, will have a low TF-IDF.

By itself, obtaining the TF-IDF characteristic allows only to rank certain terms, but not to convert the text into a numerical representation. Word Embedding technology exists to perform this action, which maps words or phrases into vectors of real numbers.

The specified technology is a set of various methods, one of which is GloVe (Global Vectors). GloVe is an algorithm for transforming unlabeled data (in our case terms) into continuous vectors to reduce dimensionality [10]. GloVe vectors are pre-trained on data from Wikipedia and Gigaword 5, so they capture the semantics of sentences quite well. It is worth noting that this algorithm is aimed specifically at texts in English, not Ukrainian.

Having clarified the key concepts, we can move on to how exactly fake news differs from real news (without reference to architectural models of neural networks). For the sake of demonstration, let's use the data set for the 2016 presidential election in the United States [11]. It contains 20015 news, of which 11941 are fake and 8074 are real. The real news in this set is data from well-known authoritative news websites, such as the New York Times, Washington Post, etc. As you can see in the figure below (see Fig. 1), they contain a lot of different information, including the title of the article, text, image, author, website, and much more. In the future, we will only use the name and text.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Unnamed: 1	author	comments	country	crawled	domain_rar_id		language	likes	main_img_tord_in_thre	participants	published	replies_cou	shares	
2	1	NA	JEREMY W.	0	US	#####	0	3	english	0	https://stat	0	0	#####	0	0
3	2	NA	STEVE EDEF	0	US	#####	0	4	english	0	https://stat	0	0	#####	0	0
4	3	NA	MAGGIE HA	0	US	#####	0	5	english	0	https://stat	0	0	#####	0	0
5	4	NA	NELSON D.	0	US	#####	0	6	english	0	https://stat	0	0	#####	0	0
6	5	NA	MAGGIE HA	0	US	#####	0	7	english	0	https://stat	0	0	#####	0	0
7	6	NA	ERIC LIPTOF	0	US	#####	0	8	english	0	https://stat	0	0	#####	0	0
8	7	NA	THE ASSO	0	US	#####	0	9	english	0	https://stat	0	0	#####	0	0
9	8	NA	SARAH MA	0	US	#####	0	10	english	0	https://stat	0	0	#####	0	0
10	9	NA	VIVIAN YEE	0	US	#####	0	11	english	0	https://stat	0	0	#####	0	0
11	10	NA	MATTHEW	0	US	#####	0	12	english	0	https://stat	0	0	#####	0	0
12	11	NA	MICHAEL C	0	US	#####	0	13	english	0	https://stat	0	0	#####	0	0
13	12	NA	ALAN RAPP	0	US	#####	0	14	english	0	https://stat	0	0	#####	0	0
14	13	NA	STEVE EDEF	0	US	#####	0	15	english	0	https://stat	0	0	#####	0	0
15	14	NA	KATIE ROGE	0	US	#####	0	16	english	0	https://stat	0	0	#####	0	0

Figure 1: Part of target dataset

After finding TF-IDF characteristics, it was found that fake news headlines often contain words like “notitle”, “IN”, “THE”, “CLINTON” and many unrelated numbers representing special symbols. Several interesting conclusions can be drawn from this. First, most fake news stories don't have headlines. These fake news stories are widely distributed in the form of tweets with several keywords and hyperlinks to news stories on social media. Secondly, in fakes, most characters are in uppercase. The goal is to grab the attention of readers, and real news has fewer capital letters and is generally written in a standard format. Third, real news contains more detailed descriptions. For example, names (Jeb Bush, Mitch McConnell, etc.) and verbs (left, claim, debate, poll, etc.). For a deeper understanding let's consider the problem of detecting fakes from the perspective of computational linguistics, psychological positioning, lexical diversity, and sentiment analysis.

The first thing to note is that fake news has fewer words on average than real news: 4,360 words versus 3,943 (84 sentences versus 69). In addition, the range of words in fake news is much larger than in real data. Despite fewer characters and sentences, the sentences themselves are shorter in fake news. This is because editors and journalists use certain newspaper norms, which include the length and choice of words, the absence of grammatical errors, etc. However, fake news is usually not based on these rules.

According to the obtained results, it can be noted that real news has fewer question marks than fake news. The reason may be that fake news is rich in rhetorical questions that are used to deliberately emphasize ideas and reinforce sentiments. In addition, it can be noted that both types of news have very few exclamation points, and therefore exclamation marks, although their number is still greater in fake news.

Particular attention should be paid to the cognitive load expressed in the words of opposition ("but", "without", "however") and denial ("no", "not"). True texts contain more objections and oppositions. This can be explained by the fact that the creator of fake news should be more specific and accurate and pay attention to appeals. This will also reduce the likelihood of self-contradiction [12].

It is also worth noting that first-person pronouns ("I" and "we") are less often used by criminals. Such a state of affairs can be connected with the need to present false information from an objective point of view. In this case, it is more appropriate to use impersonal sentences, or addresses of the second and third person plural ("you", "they"). Some psychological studies indicate that such use of pronouns is not only characteristic of fake news, but of people who tell lies in general. The reason is that in this way people shift responsibility for deception from themselves to others [13].

Lexical diversity, in general, is a measure of how many different words are used in a text, while lexical density is a measure of the proportion of lexical items (i.e., nouns, verbs, adjectives, and some adverbs) in a text. Considering the obtained data, we can note that there is more diversity in real news:

2.2e-06 versus 1.76e-06 for fake news. This can be explained by the fact that false information is usually aimed at a less educated population that does not have developed critical thinking.

The mood in real and fake news is significantly different: fake news is more negative. Several reasons can be identified that would explain this phenomenon. First, the creation of false information is aimed at inciting the population (it is worth noting that the current statement is valid for both political and other spheres of activity), accordingly, the mood of the text should have a negative color. In addition, some psychological studies indicate that this is caused by the author's inner sense of guilt [14]. Having clarified the main nuances that can affect the course of the experiment, we formalize the goal set in the introduction of the current work.

Given a set of  $m$  text news, which can be represented as follows:

$$\mathcal{A} = \{A_i^T\}_i^m, \quad (1)$$

In the task of detecting fake news, it is necessary to predict whether there are articles with fake news or not. At the same time, a set of labels indicating the veracity of the information can be presented in the following form:

$$y = \{1,0\}_i^m, \quad (2)$$

where 1 – real news, 0 – fake.

The set of functions  $X_A^T \ni A_i^T \in \mathcal{A}$  should be obtained by parsing the text of the article, e.g. using TF-IDF and Word Embedding.

In this way, the following model of formation of news labels will be embedded in the selected architectures  $\mathcal{A}$ :

$$f: \{X_i; T\}_i^m \in \mathbb{X} \rightarrow y \quad (3)$$

### 3. Mathematical representation

After obtaining a mathematical representation of the problem and reviewing the necessary data regarding the subject area, we will proceed to consider the selected architectures.

#### 3.1. Convolutional Neural Network

In general, the convolutional neural network (see Fig. 2) is a rather unique type of neural network that was primarily used to analyze graphical information. This uniqueness will be manifested in several features:

- they have three dimensions: depth, height, and width;
- nodes in each of the layers are connected only to a small part, not to all;
- the result is collapsed into a set of probabilistic estimates grouped along the depth axis;
- to determine the descriptors, the network performs several operations of collapsing and combining;
- in addition to finding descriptors, data classification takes place.

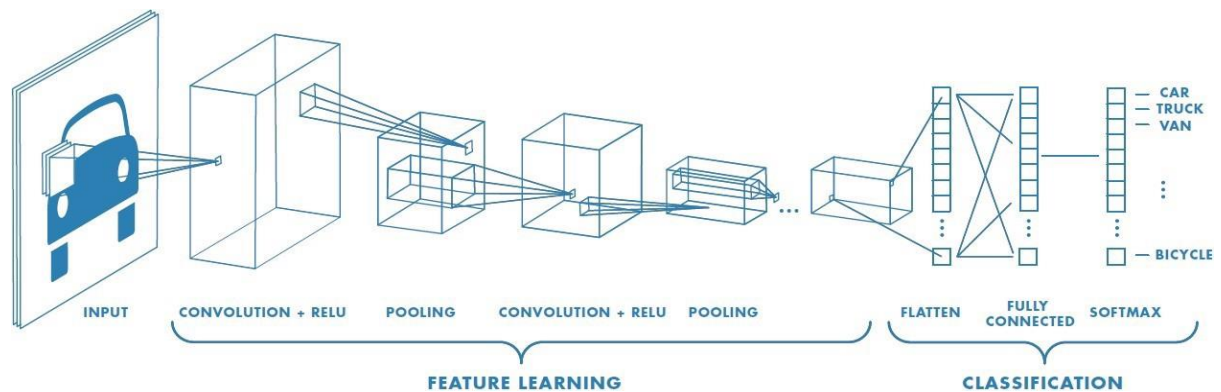


Figure 2: Overview of convolution neural network [15]

As can be seen from Figure 2, CNN consists of three types of layers: convolutional, subsampling, and perceptron. The main layer of a convolutional neural network can undoubtedly be considered the layer of convolution, the work of which is the basis of this network. Its parameters are filters that have small spatial dimensions (width and height) but pass through the entire depth of the input data volume. For example, the standard filter of the first layer of CNN can be an  $5 \times 5 \times 3$  tensor. While passing through the network, the filter is slid over the width and height of the input data, during which the scalar product between the filter entries and the input information is found. As a result, a two-dimensional activation map is formed, which provides the response of this filter at each spatial position. The number of similar activation cards is equal to the number of used filters.

After the activation maps are formed, the activation function is output to the next layer. As an example, which is often used to analyze textual information, we can cite the ReLU function:

$$f(x) = \max(0, x) \quad (4)$$

For this network, three hyperparameters are considered, which control the size of the output information. The first of them is depth. It is equivalent to the number of filters used in the network. The second is the stride with which the filter performs the pass: the larger it is, the smaller the filter is applied and the smaller the size of the output matrix. When passing the filter, there are situations when it is convenient to fill data boundaries with zeros. The extent of this padding is the final third hyperparameter. It's worth mentioning last two important parameters – kernel size and the bias.

In the case of considering the analysis of text information, the general principle of building the architecture remains unchanged, only the text vectorization process is added.

Unlike images, textual information is one-dimensional, so the number of convolutions is 1.

Mathematically, in this case, the key layer of the CNN linear operation can be represented as some linear layer  $W \in \mathbb{R}^{(k \cdot d) \times m}$  for a window  $(x_i, \dots, x_{i+k-1}) \in \mathbb{R}^{k \cdot d}$  of size  $k$ . At the same time, the operation itself takes as input a concatenation of vectors:

$$u_i = [x_i, \dots, x_{i+k-1}] \in \mathbb{R}^{k \cdot d} \quad (5)$$

and performs multiplication by the convolution matrix:

$$F_i = u_i \times W \quad (6)$$

Regarding the key parameters, the kernel size usually ranges from 2 to 5. It is not advisable to choose Stride greater than 3 when examining the text, as this will affect the overall quality of the information analysis, and considering not the subject area, this parameter is very important. Depth is 1, so adding zeros is only necessary if there is a non-unit step. Regarding the bias parameter, it is usually not used, although such a possibility is provided.

After the convolution is obtained, the union layer aggregates the features in a certain region. There are several methods of pooling, however, when considering text information, the most popular is max-pooling, which selects the maximum value of each filter, and it will be used later.

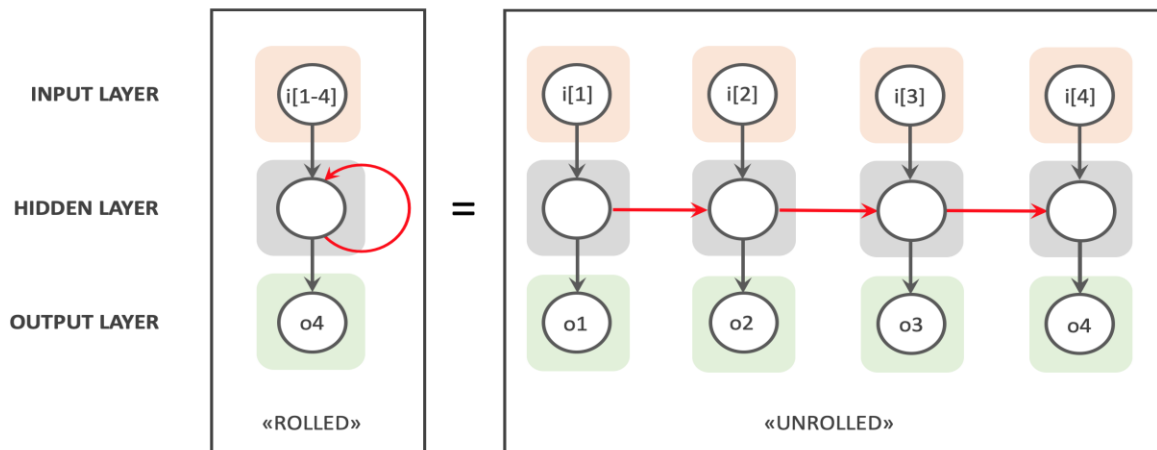
Since news texts are sometimes quite long, it is necessary to apply many layers to them. Unfortunately, in this case, there may be a problem with the propagation of gradients from top to bottom through the deep network. To avoid this, you can use residual connections or a more complex option – trunk connections.

After the unification, the process of obtaining a probability distribution, which is usual for all neural networks, takes place.

## 3.2. Recurrent Neural Network

Having considered the architecture of convolutional networks, let's move on to recurrent networks (see Fig. 3). Among their general features, it is worth highlighting:

- a cyclic connection between the layers;
- a large number of modifications depending on the amount of data;
- a two-dimensionality;
- a short-term memory to consider previous data when creating an output.



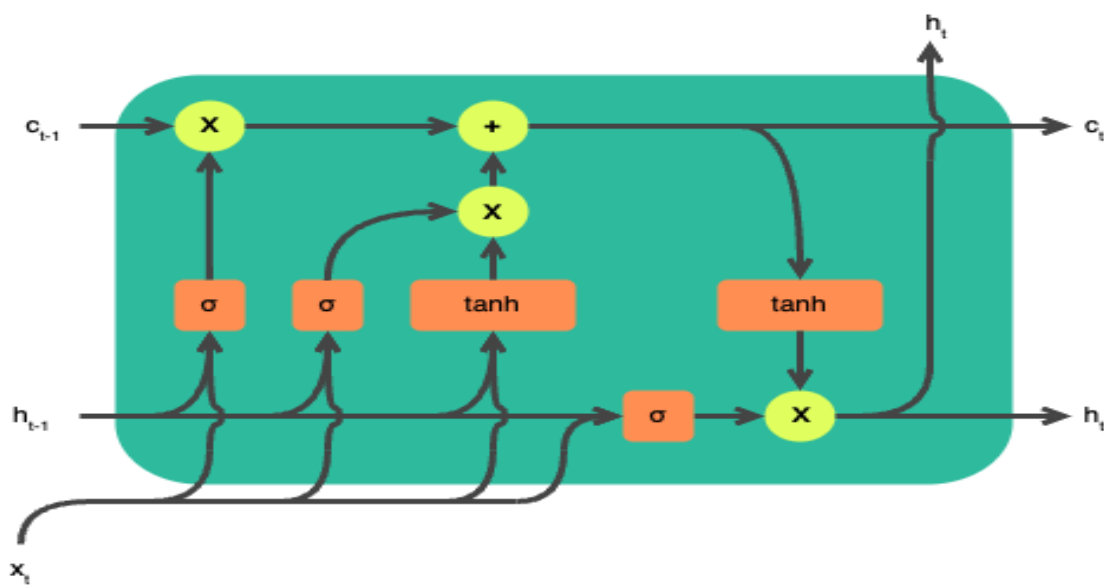
**Figure 3:** Overview of recurrent neural network [16]

Figure 3 depicts the recurrent boundary at each point in time, showing the presence of three layers: input, hidden (where the main processing is done), and output. The learning process occurs by back-expansion in time when new weights are obtained both as a result of current processing and due to previous values. This process is formally a manifestation of feedback, which was noted when considering CNN.

The input data layer performs the initial processing of the received text information. It can include tokenization, lemmatization, embedding words, etc. [17]. The hidden layer contains a rule according to which input data is processed. And the output layer converts the result into the required format.

The process of finding weight coefficients, which takes place in the so-called "hidden" layer, is related to finding gradients based on input data. This feature, in turn, can lead to two problems – a vanishing gradient and an explosive gradient. The first is related to large amounts of data or incorrect initial configuration, the gradients determining the weighting coefficients can go to 0. As a result, the network will stop the learning process. The second is the opposite of the first problem.

To overcome these shortcomings, which can have a significant impact when considering fake news, in the future, we will use not the classic architecture of the network, but its modification with support for both short-term and long-term memory – LSTM. Such a move will allow analyzing text information of uncertain volume more precisely. Below (see Fig. 4) is a schematic representation of the operation of one of the steps of finding the weighting coefficients:



**Figure 4:** Overview of LSTM [18]



Let's take a closer look at the indicated figure, and for a simplified understanding we will use the original names of the elements.

The first stage of the LSTM hidden layer is the Forget Box. Here, the two input weights are simply added and multiplied by the activation function  $\sigma$  so that the result is between 0 and 1. Mathematically, this can be represented as follows:

$$F = \sigma(x_t + h_{t-1}) \quad (7)$$

where  $x_t$  – input data,  $h_{t-1}$  – preliminary output data,  $\sigma$  – activation function.

As we can see, the result of this addition is multiplied by the data from memory. That is why the stage was named oblivion. If the input value turns out to be 0, then when multiplying by a memory cell, we can successfully delete the value.

The next stage is the Input Box, which is responsible for storing data in memory. The output range of the sigma function is between 0 and 1, i.e. it is always positive. In this case, all values obtained as a result of Forget Box will be written to memory. That's why the result is multiplied by the tangent, which has an output range of -1 to 1. This way, some of the data is filtered out as insignificant. Mathematically, the work of this stage can be presented as follows:

$$I = \sigma(x_t + h_{t-1}) \quad G = \tanh(x_t + h_{t-1}) \quad (8)$$

As a result of the operation of the two indicated stages, the state of memory  $C$  is formed, below  $c_{t-1}$  denotes the previous state of memory:

$$C = c_{t-1}F + IG \quad (9)$$

The received state of the memory, like the input data and the previous output, serve as the basis for the formation of a new output state  $O$ , which is the final stage of the execution of the model step:

$$O = \tan(C) \times \sigma(x_t + h_{t-1}) \quad (10)$$

Now we can move on to setting up an experiment that will be conducted to test the effectiveness of each type of architecture.

## 4. Experimental environment

The experimental environment in the current study is the following set of characteristics:

- efficiency function;
- rule for comparing the efficiency of two models;
- experimental plan.

We will gradually determine each of the specified characteristics.

### 4.1. Efficiency function

The efficiency of a certain model ( $E$ ) will be determined as follows:

$$E = f(\vartheta, MSE, \nu) \quad (11)$$

where  $\vartheta$  – training speed;  $MSE$  – mean square error of the forecast;  $\nu$  – data volume.

The performance indicator can be determined using third-party modules or software after implementing the models. For example, the time module or software systems Postman, JMeter.

To determine the accuracy of the forecast, we will use special samples with data on the Russian invasion of Ukraine [19] and the election of the president of the United States in 2020 [20]. The specified data will be divided according to the Pareto principle into two groups in the ratio of 80 to 20. 80% will be used as input data for forecasting. 20% will instead serve as the real value. After classification, to find the root mean square error, it is enough to use the following formula:

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (12)$$

where  $N$  – number of classified values;  $y_i$  – real value;  $\hat{y}_i$  – classified value.

As the volume of data, we will consider the time spent by the user to prepare the model for use. At the same time, to correctly compare the two models, the inverse time indicator will be considered.

## 4.2. Efficiency comparison rule

To compare the efficiency of the two models, we introduce the variable  $C$ , which is determined by the following formula:

$$C_{AB} = \sum \text{normalize}\left(\frac{m_A}{m_B}\right), \quad (13)$$

where  $m_A$  – metric value for model  $A$ ;  $m_B$  – metric value for model  $B$ .

After obtaining the results for each model, it is possible, using a fuzzy parameter, to determine which model is more effective:

- $C_{AB} \geq 3.5$ : model  $A$  is more efficient than  $B$ ;
- $C_{AB} \leq 2.5$ : model  $A$  is less efficient than  $B$ ;
- $2.5 < C_{AB} < 3.5$ : it's impossible to determine whether there is one model more efficient than another.

## 4.3. Experimental plan

Numerical data must be collected for all metrics of each model, so a controlled experiment method is used. As a result, a stable and permanent execution environment was chosen – a physical device based on Ubuntu, the technical specifications of which are given below:

- CPU: Intel Core i5-1135G7;
- RAM: 16 Gb;
- SSD: 512 Gb;
- VRAM: 4 Gb;
- OS: Ubuntu 21.04.

To determine the performance, each model is implemented using Python 3. To avoid problems with manual testing and simplify the debugging process, it was decided to use the Jupyter software environment and the time module for performance evaluation.

As mentioned earlier, two test samples are used to determine accuracy. This data is divided into two parts in the ratio of 80/20. The accuracy result obtained as the ratio of the obtained class and the real one for two samples is aggregated.

For a better understanding, we present a fragment of the data for each sample. Let's start with data on the invasion of the Russian Federation into the territory of Ukraine (see Fig. 5):

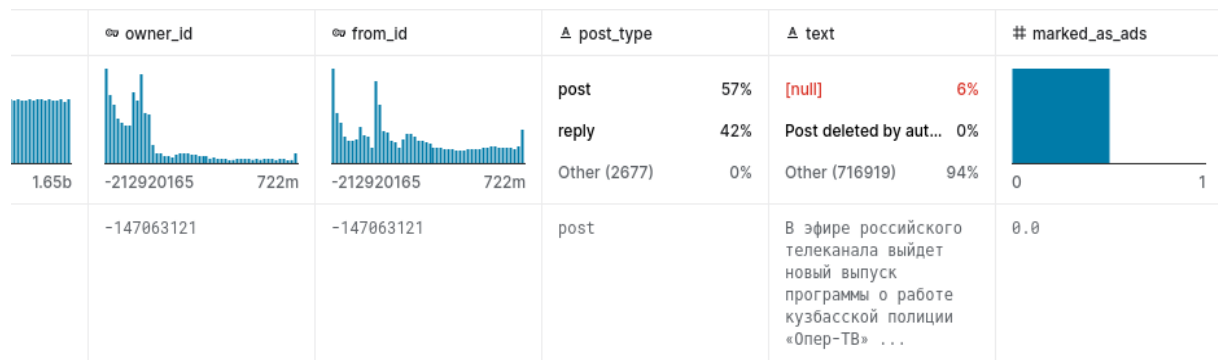


Figure 5: Part of first dataset

This dataset is described by the following fields:

- id: unique id for a news article;
- date: news publication date;
- text: the text of the article; could be incomplete;
- owner-id: id of the author of the post;



- from-id: the identifier of the author of the message;
- post-type: since the data is information from social networks, the indicator indicates whether it is a reply or an actual post;
- attachments: additional images for news;
- marked-as-ads: a label that marks the article is unreliable: 1 – unreliable, 0 – reliable.

The data on the presidential elections in the USA differ in content and represent the following information (see Fig. 6):

id	title	author	text	label
unique id for a news article	the title of a news article	author of the news article	the text of the article; could be incomplete	a label that marks the article as potentially unreliable 1: unreliable 0: reliable
	[null] 3% The Dark Agenda B... 0% Other (20237) 97%	nan 9% Pam Key 1% Other (18600) 89%	<b>20387</b> unique values	
0	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It	Darrell Lucus	House Dem Aide: We Didn't Even See Comey's Letter Until Jason Chaffetz Tweeted It By Darrell Lucus o...	1

Figure 6: Part of second dataset

This dataset is described by the following fields:

- id: unique id for a news article;
- title: the title of a news article;
- author: author of the news article;
- text: the text of the article; could be incomplete;
- label: a label that marks the article as potentially unreliable: 1 – unreliable; 0 – reliable.

First, it is necessary to remove redundant fields from the datasets. In fact, the text and the fake marker are the main ones in the current study.

It is worth noting that the number of fake news for each of the datasets is about 50%. In addition, each sample will first be divided into test and training samples in the ratio of 20% to 80%, and then the training sample will be dynamically divided into validation and actual training samples. Each training epoch, validation will take place on 10% of the total training sample.

The final indicator for analysis is the speed of data preparation. In this case, the indicator depends solely on the person performing the test.

The following errors and uncertainties can be determined for this experiment:

- when checking the speed of work, errors and errors related to the time execution time estimation module and the Jupyter software environment should be expected;
- when checking the accuracy of the models, problems may arise with the data used as a test sample, as they directly affect the obtained result;
- when checking the speed of data preparation for analysis, two main problems can be identified: the human factor and the error of the time measurement tool.

## 5. Models implementation

First, all non-alphanumeric characters (such as numbers, commas, periods, and other punctuation marks) are removed from the text using the re library, which provides access to regular expressions. After cleaning, the text goes to the processing function, its code can be seen below (see Fig. 7):

```

def process_text(text: str, language: str) -> str:
    stemmer = PorterStemmer()
    lemmatizer = WordNetLemmatizer()
    stop_words = set(stopwords.words(
        get_language_name(language)
    ))
    sentences = nltk.sent_tokenize(text)
    sentences_processed = []
    for sentence in sentences:
        words_processed = []
        words = nltk.word_tokenize(sentence)
        words = [word for word in words if word not in stop_words]
        for word in words:
            word = stemmer.stem(word)
            word = lemmatizer.lemmatize(word)
            words_processed.append(word)
        words = [word for word in words_processed if word not in stop_words]
        sentences_processed.append(" ".join(words))
    text_processed = " ".join(sentences_processed)
    return text_processed

```

**Figure 7:** Part of code for text processing

Central to this feature are the methods provided by the nltk library, which allows natural language processing based on built-in word corpora.

As a first step it is necessary to remove from the set those words that do not carry an informational load (for example, "and", "or", etc.). These words will interfere with the correct analysis.

The next step is to level the linguistic variability generated using morphemes. For this, the operation of stemming (from English stemming) is used – the process of reducing a word to its base. For example, the words “eating” and “eaten” will be replaced by the word “eat”. After this processing, the array of words is reduced to a set of bases.

The class built into the nltk library, which performs the specified process, does not always work correctly, therefore, to improve the result, it was decided to combine stemming with another operation – lemmatization (from English lemmatisation). It allows you to bring the word form to the lemma (normal dictionary form). For example, when processing the words "good" and "better" will have different bases, but these words have the same lemma – "good". After carrying out the specified actions, the set of words needs another check for non-informative content.

After processing the text, a dictionary is created that is necessary for the correct definition of TF-IDF characteristics. In order to take into account the emotional coloring of words, the next step is to find a frequency-polar characteristic for each word, by multiplying the TF-IDF and polarity indicators. The latter can be obtained by using the SentimentIntensityAnalyzer class found in the vader module of the nltk library.

After that, it is necessary to sum up the obtained multiplications and get a single result for each news entry.

It is worth noting that in the two cases the languages of the news are different (English for the election sample and Russian for the war sample), so it is necessary to use different submodules for stop words..

## 6. Experiment results

Before directly comparing the efficiency of the two algorithms, we note that because of the analysis of the ratio of loss with the number of epochs, it was established that the optimal number of epochs for CNN is 3 epochs, and for LSTM – 4 epochs.

Let's move on to the definition of efficiency parameters and start with training speed measurements using the time library. The results are shown in the table below (see Table 1):

**Table 1**  
Training speed of neural networks

CNN	RNN
305 s	325 s
294 s	311 s
311 s	336 s
290 s	301 s
301 s	316 s
303 s	319 s
307 s	334 s
297 s	310 s
303 s	323 s
299 s	317 s

Let's find the average value for each of the algorithms. For CNN, we have a time value of about 301 s, for RNN – ~319 s. Let's move on to the next indicator – classification accuracy. As mentioned above, the accuracy of the forecast is measured using the MSE indicator. Below we present the accuracy of the obtained results for both samples (see Table 2):

**Table 2**  
Accuracy

Name	CNN	RNN
War	93%	94%
Election	94%	96%

The last indicator is the time of data preparation of architectural models. The measurements results for this indicator are shown in the table below (see Table 3):

**Table 3**  
Time of data preparation

CNN	RNN
419 s	421 s
382 s	378 s
359 s	356 s
395 s	398 s
401 s	403 s

Let's find the average value of the indicator for each of the algorithms. We have ~391 s for CNN and ~391 s for RNN.

Having found the appropriate metrics for comparing the efficiency of the two algorithms, we will move on to finding the CAB indicator. We will consider CNN as model A, and RNN as model B. We have CAB ~ \$2.83. This allows us to state the higher efficiency of RNN versus CNN, although this difference is not significant. While giving gains in training speed, CNN loses in prediction accuracy. In the case of the implementation of algorithms within the framework of a cloud platform, which makes it possible to significantly accelerate the speed of work, the resulting difference in speed between the models will be insignificant. However, it is worth noting that accuracy may vary depending on the data and its volume.

## 7. Conclusion

The current work aimed to investigate the effectiveness of using neural network architectures such as CNN and RNN to detect fake news. For this purpose, the subject area associated with false information was analysed and key patterns characterizing similar data were identified. In addition, the theoretical background of both types of models was analysed. During the investigation, it was found that standard models cannot be effectively applied for text classification, so it was decided to keep in mind the following provisions for further consideration:

- CNN should be understood as a convolutional neural network with a single-layer convolution and the appropriate adjustment of all its parameters;
- under RNN, understand an LSTM network configured for text analysis, which has both long-term and short-term memory.

The next step was to define the experimental environment. As test samples, it was decided to choose data related to the Russian invasion of the territory of Ukraine and the presidential elections in the United States in 2020.

As a result, it was discovered that the CNN model works faster than RNN on the presented data, but gives a less accurate classification result. Given the proposed efficiency comparison rule, the stated probabilities of errors of various kinds, and ways to overcome the disparity between algorithms, the resulting efficiency gains can be considered insignificant.

This conclusion corresponds to the world scientific practice, which recommends that during the analysis of textual information, especially, in the presence of two classes (fake and non-fake data in the current case), use any of the proposed models, or their congregation in the case of checking images for authenticity. This, in turn, proves the feasibility of using similar algorithms for news filtering systems, for example as part of a social network or search engine.

As further goals for deepening the research, it is proposed to consider more broadly the possibilities of combining recurrent and convolutional networks.

## 8. References

- [1] "fake news." Cambridge Dictionary. <https://dictionary.cambridge.org/dictionary/english/fake-news> (accessed Feb. 8, 2023).
- [2] Polygrph. "The Infodemic: Did China Deliver a COVID-19 Vaccine to Africa?" Voice of America. [https://www.voanews.com/a/covid-19-pandemic\\_infodemic-did-china-deliver-covid-19-vaccine-africa/6187618.html](https://www.voanews.com/a/covid-19-pandemic_infodemic-did-china-deliver-covid-19-vaccine-africa/6187618.html) (accessed Feb. 8, 2023).
- [3] L. Chinchilla. "How Does the Age of Fake News Impact Democracy in the Developing World?" Kofi Annan Foundation. <https://www.kofiannanfoundation.org/supporting-democracy-and-elections-with-integrity/annan-commission/post-truth-politics-afflicts-the-global-south-too/> (accessed Feb. 8, 2023).
- [4] D. Avelar. "WhatsApp fake news during Brazil election 'favoured Bolsonaro'." The Guardian. <https://www.theguardian.com/world/2019/oct/30/whatsapp-fake-news-brazil-election-favoured-jair-bolsonaro-analysis-suggests> (accessed Feb. 8, 2023).
- [5] K. Wesolowski. "Fake news further fogs Russia's war on Ukraine." Deutsche Welle. <https://www.dw.com/en/fact-check-fake-news-thrives-amid-russia-ukraine-war/a-61477502> (accessed Feb. 8, 2023).
- [6] I. Afanasieva, N. Golian, O. Hnatenko, Y. Daniil, and K. Onyshchenko, "Data exchange model in the Internet of Things concept," *Telecommunications and Radio Engineering*, vol. 10, no. 78, pp. 869–878, 2019.
- [7] A. Blocman. "Laws to combat manipulation of information finally adopted." IRIS Merlin. <https://merlin.obs.coe.int/article/8446> (accessed Feb. 8, 2023).
- [8] V. Golian, N. Golian, I. Afanasieva, K. Halchenko, K. Onyshchenko, and Z. Dudar, "Study of Methods for Determining Types and Measuring of Agricultural Crops due to Satellite Images," in *32nd International Scientific Symposium Metrology and Metrology Assurance*, Sozopol, Bulgaria, Sep. 7–11, 2022.

- [9] P. S. Reddy, D. E. Roy, P. Manoj, M. Keerthana, and P. V. Tijare, "A Study on Fake News Detection Using Naïve Bayes, SVM, Neural Networks and LSTM," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 06, pp. 942–947, 2019.
- [10] J. Pennington, R. Socher, and C. D. Manning. "GloVe: Global Vectors for Word Representation." Stanford Edu. <https://nlp.stanford.edu/projects/glove/> (accessed Feb. 16, 2023).
- [11] M. Risdal. "Getting Real about Fake News." kaggle. <https://www.kaggle.com/datasets/mrisdal/fake-news> (accessed Feb. 16, 2023).
- [12] A. Dey, R. Z. Rafi, S. Hasan Parash, S. K. Arko and A. Chakrabarty, "Fake News Pattern Recognition using Linguistic Analysis," *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, Kitakyushu, Japan, 2018, pp. 305-309.
- [13] D. K. Sharma, P. Shrivastava and S. Garg, "Utilizing Word Embedding and Linguistic Features for Fake News Detection," *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2022, pp. 844-848.
- [14] X. Jose, S. D. M. Kumar and P. Chandran, "Characterization, Classification and Detection of Fake News in Online Social Media Networks," *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, Hassan, India, 2021, pp. 759-765.
- [15] S. Saha. "A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way." Towards Data Science. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (accessed Feb. 23, 2023).
- [16] A. Trujillo. "LSTM Neural Network for Time Series Forecasting." RPubs. <https://rpubs.com/atrujill/717267> (accessed Feb. 23, 2023).
- [17] IBM Cloud Education. "Natural Language Processing (NLP)." IBM Cloud Learn Hub. <https://www.ibm.com/cloud/learn/natural-language-processing> (accessed Feb. 20, 2023).
- [18] M. West. "Explaining Recurrent Neural Networks." bouvet. <https://www.bouvet.no/bouvet-deler/explaining-recurrent-neural-networks> (accessed Feb. 23, 2023).
- [19] DS-Pr1nce. "War in Ukraine: Russian social network discussions." kaggle. <https://www.kaggle.com/datasets/ustyk5/war-in-ukraine-russian-social-network-discussions> (accessed Feb. 23, 2023).
- [20] R. Dedhia. "Fake News." kaggle. <https://www.kaggle.com/datasets/ronikdedhia/fake-news> (accessed Feb. 23, 2023).