RESEARCH ARTICLE

Methods of Identifying Fake News in Social Networks

Mariia Nazarkevych[1], Victoria Vysotska[1], Vladyslav Liakh[1], Yelyzaveta Leheza[1], Nazar Naconechnyi[1]

1 Lviv Polytechnic National University, Ukraine

Funding: The research was carried out with the grant support of the National Research Fund of Ukraine "Information system development for automatic detection of misinformation sources and inauthentic behaviour of chat users ", project registration number 187/0012 from 1/08/2024 (2023.04/0012).

Potential competing interests: No potential competing interests to declare.

Abstract

False information is present in media news, in the information space, because anyone can write news. In addition, the presence of martial law in Ukraine provokes a hybrid war in the information space, and it is constantly necessary to counter the threats. To date, there is manipulation of public disinformation opinion in media news and social networks. The availability of effective methods of identifying fake news and countermeasures is aimed at this study. We will use machine learning methods to detect disinformation. A dataset for detecting fake news has been developed. The most frequently used words in fake news, collected from August to November 2024, were studied.

Introduction

Social networks are not only a communication tool and a source of information but also a space for spreading various information threats. On the other hand, informational threats negatively impact the opinion of ordinary citizens, who are unprotected from the actions of intruders.

The most relevant are disinformation, fakes, manipulation of public opinion, and publications by bots and trolls. Let's look at these threats and analyse their impact on domestic cyberspace.

Misinformation is deliberately distorted or false information[1] disseminated to confuse people or influence their actions, emotions, or attitudes. Disinformation can manipulate public sentiment, discredit individuals or organisations, and weaken state structures. It is most often distributed through social networks, news portals or messengers. The potential of media literacy to respond to misinformation was analysed by R. Hobbs[2], who conducted an applied study on the right to teach responses to conspiracy theories. Subsequently, research in the international field of media literacy appeared to assess its ability to resist fake news[3]. The spread of fake news or distorted information is one of the biggest threats in social media. For example, much misinformation manipulates public opinion during war or political crises. Fakes are used to demoralise society and to increase panic. Most often, fakes are created through news, images or even videos. The primary purpose of a fake is to deceive or provoke an emotional response. Fakes can reach a vast audience in a short time.

Dataset formation

One thousand four hundred news stories that were distributed in Ukrainian content from September 2024 to October 2024 were analysed (Fig.1). The dataset was built in such a way that it recorded information that was published on the Internet on sites that distribute news, on social networks Telegram, Facebook, Instagram.

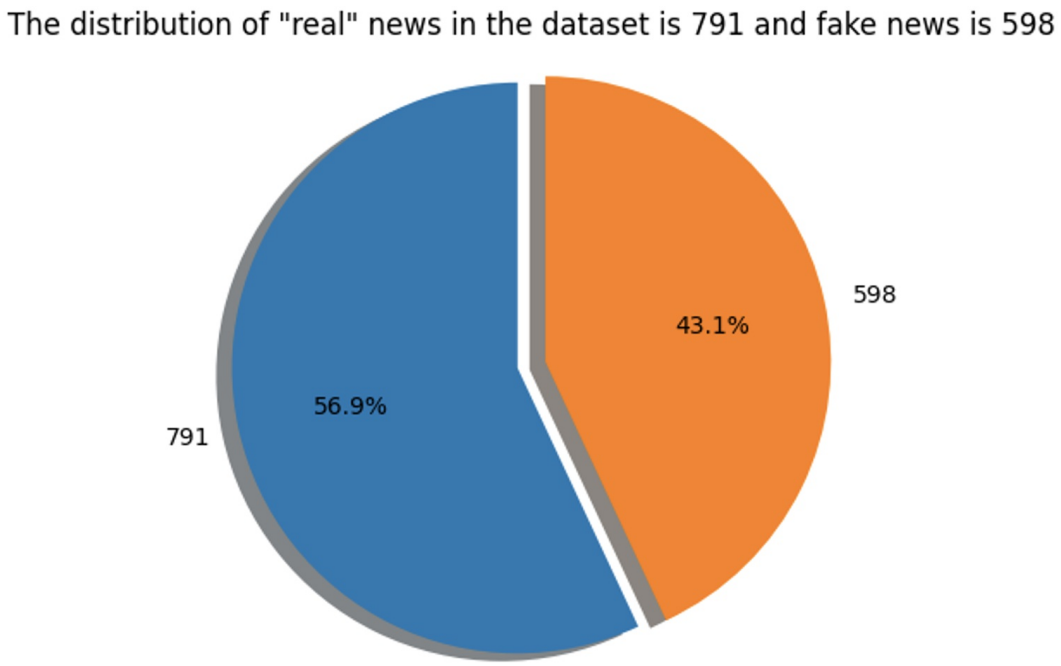

Fig.1. The distribution of real news in the dataset

Distribution in the dataset of news Telegram, Facebook, news on sites, Instagram, Twitter, Vkontakte, and Yandex is shown in Fig.2, Fig.3, Fig.4, and Fig.5.

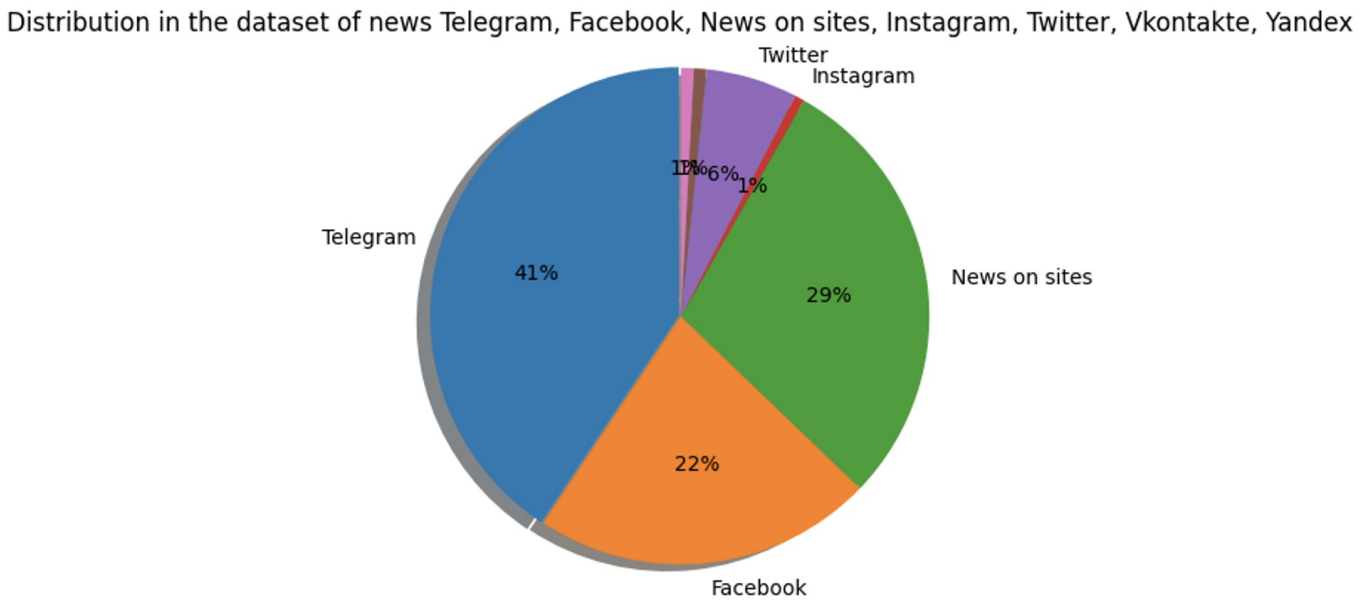

Fig.2. Distribution of messages and fake messages in Telegram, Facebook, news on the sites, Instagram, Twitter, Vkontakte, Yandex

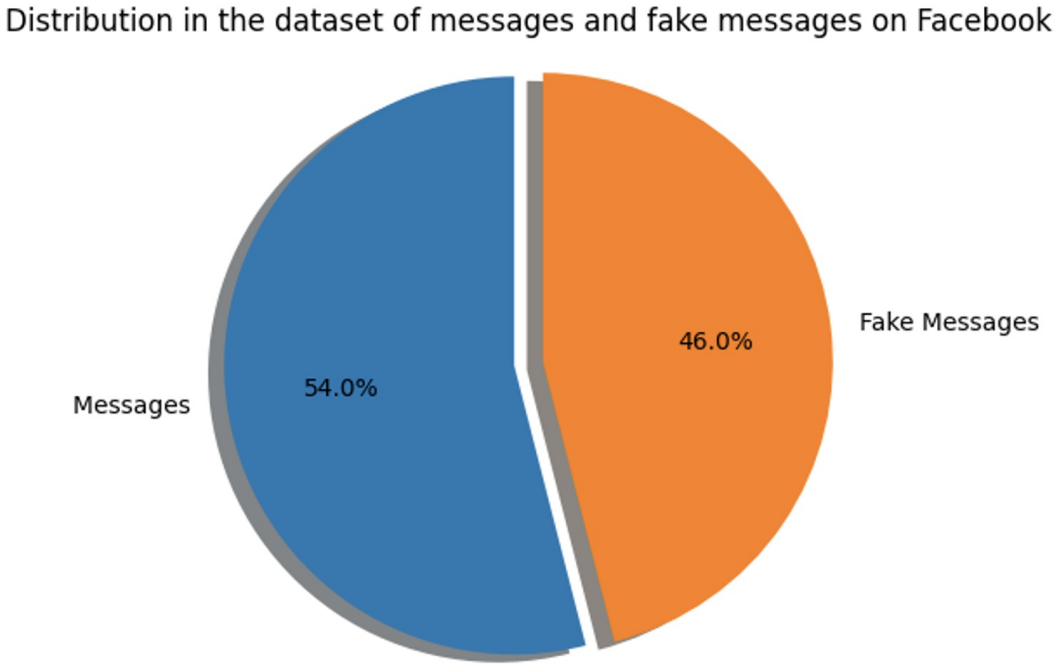

Fig.3. Distribution of messages and fake messages on Facebook

In this way, the dataset was formed because there were equal numbers of "true" and fake messages.

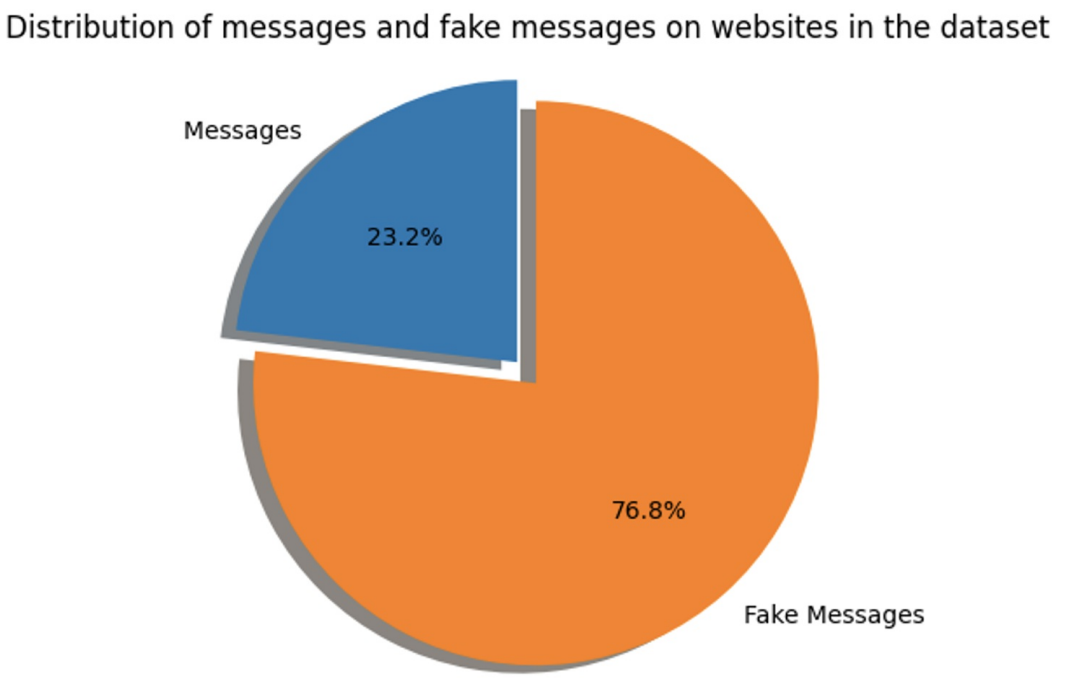

Fig.4. Distribution of messages and fake messages in news on the sites

Seven thousand likes were received by a fake Telegram message with the following content: "The Russian army destroyed the point of temporary deployment of Ukrainian fighters in Sumy – explosions can still be heard in the city. Today, the Air Force of the Russian Federation did not stop at what was achieved in Poltava and continued in the same spirit - yes, the Russian army struck the point of temporary deployment of the Armed Forces in Sumy. As is customary with the enemy, the location was in the building of the local university. According to preliminary information, the reserves were located there, which were probably supposed to go to the slaughter for Russian weapons in the Kursk border area.

A little less, 6.7 thousand. Likes were received by a fake message on Twitter with the following content: "NEWSFLASH: "If I wanted to see a drug addict, I'd just take a walk around Tijuana," Mexico's elected president explained why she declined Zelensky's invitation to visit Kyiv."

Third place, from 5 thousand likes, is taken by a fake message on Twitter with the following content.

> *"AZOV committed atrocities. They cut off the heads of old people and tortured children. But the biggest horror is the mercenaries. They hung the flag of France in the neighbouring village" - Rashistka from Korenevo*

Among the actual messages, about 39,000 people liked the message with the following content: "15 Azov servicemen have been returned from captivity!" published in Telegram.

Average readers also like the news: "The national futsal team of Ukraine defeated the Netherlands and reached the quarterfinals of the 2024 World Cup. In the 1/4 finals, "blue and yellow" will play against the winner of the Spain-Venezuela match." more than 3,500 readers liked it.

People also like the news "Kyiv. Okhmatdyt Hospital. The largest children's hospital in Ukraine. Children and adults are pulled out from under the rubble. Ruined resuscitation. The new hospital building and half of the old one have been destroyed. The situation is tough." 3.8 thousand people liked her on Facebook.

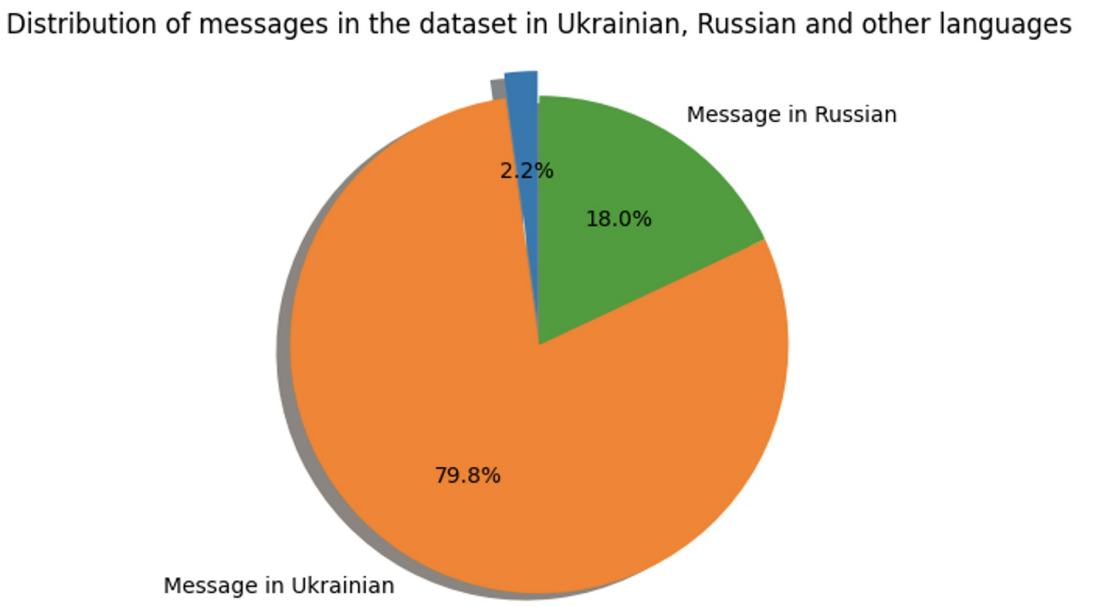

Fig.5. Distribution of messages in languages Ukrainian, Russian and other

Among the fake news, the words "warrior", "satisfied", "horror", "shock", "unbelievable", "rashishtka" were most often found.

Model training

In the first stage, training data is formed. To train the model, we need samples of goods with already known HS codes and their descriptions. If you don't have codes for all products, some products will only be used for prediction (classification) when the model is ready. We divide the data into training and test sets - 70-80% for training and 20-30% for testing. We use the training set to train the SVM model. In the next step, the data is cleaned and standardised for further use in the algorithms. The product description undergoes linguistic processing to highlight critical characteristics. By tokenisation, we divide the text into separate words or phrases. Text conversion into numerical vectors occurs using the TF-IDF (Term Frequency-Inverse Document Frequency) or Word2Vec methods.

Next, we train the model on those data where there are appropriate HS codes, and it will be able to use this information to predict the codes on new goods.

HS or HTS codes for new products are assigned after model training. Use a model to predict HS codes for goods for which a description is available, but the code is unknown. To assign HS codes, goods already described with known codes are used to train the classification model.

A program for detecting fakes in a dataset can use machine learning and natural language processing (NLP) methods[4][5]. The basic idea is to train a model on a dataset where true and fake news (or other types of information) are already marked. The model can then be used to classify new data.

Machines are great at processing structured data. When it comes to processing free-form text, machines have a hard time. Natural Language Processing NLP aims to develop algorithms that allow computers to recognise free text. The sheer number of possible variations is one of the biggest challenges in natural language processing. Context is vital for understanding the meaning of specific sentences.

```
# 1. #Download dataset
Fafes. xlsx

file_path = './Fakes-29.xlsx'

df = pd.read_excel(file_path)
```

1. Python programming was selected to implement the task. All new items were collected from the dataset, and the column under the name “mark” is equal to zero for further analysis. At the end of the analysis, the following will be identified:

- Suspicious words. These are the words that are most often used when writing fake news.
- The language in which fake news is most often written.
- The source of literature is the one who publishes such news most often.
- Confidence that the novelty is fake when guessing words, words and phrases.

We analyse hundreds of data relations based on which dataset compositions:

```
# 2. Statistics by language

valid_languages = ['ukrainian', 'urk', 'russian','rus','english', 'french', 'rus', 'ros', 'ukrainian', 'ukr'] # Valid values of languages

df['Language'] = df['Language'].str.lower() # Lower the value for correct counting

language_counts = df['Language'].value_counts(normalize=True) * 100 # Percentage ratio

language_counts = language_counts[language_counts.index.isin(valid_languages)] # Leave only valid languages

print("Percentage of news by language:")

print(language_counts)
```

The Ukrainian language is implemented in the data set - 51%

Russian - 30%

English - 4%

Other languages - 15%

We analyse the number of new products at the right time.

```
# 3. Convert all data to datetime format

# Let's try to convert string values to time format

df['time'] = pd.to_datetime(df['time'], errors='coerce',
format='%H:%M:%S')

# Let's check which lines could not be converted

invalid_times = df[df['times'].isna()]

if not invalid_times.empty:

 print(f"Number of invalid time data: {invalid_times.shape[0]}")

# We extract an hour for further statistics

df['Hour'] = df['time'].dt.hour

# We determine the time of day based on the hour

def get_time_of_day(hour):

 if 6 <= hour < 12:

 return 'Morning'

 elif 12 <= hour < 18:

 return 'Day'

 elif 18 <= hour < 24:

 return 'Evening'

 otherwise:

 return 'Night'

# We use the function to determine the time of day

df['Time of day'] = df['Hour'].apply(get_time_of_day)

# We count the number of news by time of day

time_of_day_counts = df['Time of day'].value_counts()

print("Distribution of news by time of day:")

print(time_of_day_counts)
```

Day 353

Evening 305

Morning 222

Night 177

We filter the entries, and new ones are marked with a 0 mark, which means they are fake. A new DataFrame fake_news is created, which will replace the fake news. Three columns of fake news are collected – texts and movies.

```
# 4. # Filter records where "label" == 0 (fake news)

fake_news = data[data['label'] == 0]

# We receive texts, languages and sources of fake news

texts = fake_news['message text'].fillna("").astype(str) # Replace missing values

languages = fake_news['Language']

sources = fake_news['Source']
```

It is possible to predict the frequencies of words that appear to be suspicious words (those that move more often beyond the middle). The script combines all the texts of fake news in one row, breaks them down into words, and converts them to lowercase. For additional help, the frequency of the skin word is monitored. Then, the average frequency of words is calculated, and words that occur more often than the average are added to the list of suspicious words.

```
#5. Analysis of words: combine texts into one line and count the frequency of words

combined_text = " ".join(texts) # All texts in one line

words = re.findall(r'\b\w+\b', combined_text.lower()) # Tokenization of words (break into words)

word_counts = Counter(words) # Count the frequency of words

# Selection of words that occur frequently (above average level)

average_word_frequency = sum(word_counts.values()) / len(word_counts)

suspicious_words = {word: count for word, count in word_counts.items() if count > average_word_frequency}
```

After selecting the data, the fake_probability_and_statistics function calculates the likelihood of a novelty being fake by analysing text, language and news. Vaughn calculates suspicious words in the text, phrases for language and language, and then combines these factors with words: words (50%), language (25%), and language (25%). The result is a counterfeit virality of the fakeness of the new product.

```
#6. Function for calculating the probability of fakeness and statistics
```

```
def fake_probability_and_statistics(text, language, source):
 # Text analysis: tokenisation and counting of "suspicious" words
 words_in_text = re.findall(r'\b\w+\b', text.lower())
 total_word_count = len(words_in_text)
 # Probability by word: the ratio of the number of "fake" words to the total number of words
 fake_words_in_text = [word for word in words_in_text if word in suspicious_words]
 fake_word_count = len(fake_words_in_text)
 # We determine the probability based on suspicious words
 word_based_probability = (fake_word_count / total_word_count) if total_word_count > 0 else 0
 # Probability based on language: frequency of fake news for that language
 language_risk = language_distribution.get(language, 0) / len(languages) if len(languages) > 0 else
0
 # Probability based on source: frequency of fake news for this source
 source_risk = source_distribution.get(source, 0) / len(sources) if len(sources) > 0 else 0
 # Total probability of fakeness (weights: words — 50%, language — 25%, source — 25%)
 total_probability = word_based_probability * 0.5 + language_risk * 0.25 + source_risk * 0.25
 return {
 "total_probability": total_probability,
 "words_stats": {
 "fake_word_count": fake_word_count,
 "fake_words_in_text": fake_words_in_text # Suspicious words in the text
 },
 "language_stats": {
 "language": language,
 "language_risk": language_risk
 },
 "source_stats": {
 "source": source,
 "source_risk": source_risk
 }
 }
```

When working with the text, you must break it into smaller parts for analysis. Tokenisation breaks the input text into more minor elements, such as words or sentences. These elements are called tokens.

Let's move on to the analysis. The loop goes through all the fake news, extracts its text, language, and text, calls the fake_probability_and_statistics function to expand the probability of fakeness, and saves the results in the statistics list.

```
#7. Analysis for each message

statistics = []

for i in range(len(fake_news)):

 text = texts.iloc[i]

 language = languages.iloc[i]

 source = sources.iloc[i]

 # We get probability and statistics

 stat = fake_probability_and_statistics(text, language, source)

 statistics.append(stat)
```

Finally, the max() function is used to assign the index of the new product with the highest probability of fakeness, equal to the value of the total_probability of all new products, after which data about this new product is saved.

```
#8. Finding the maximum probability of fakeness

max_probability_index = max(range(len(statistics)), key=lambda i: statistics[i]['total_probability'])

max_stat = statistics[max_probability_index]

# Extraction of unique suspicious words from news text with maximum probability

unique_suspicious_words = set(max_stat['words_stats']['fake_words_in_text'])

# Output results for the most likely fake news

print(f"\This news has the highest probability of being fake with the following characteristics:"):")

print(f"Suspicious words: {', '.join(unique_suspicious_words)}")

print(f"Language: {max_stat['language_stats']['language']}")

print(f"Source: {max_stat['source_stats']['source']}")

# Calculation of the probability of fakeness for the mentioned words, languages and sources

combined_probability = max_stat['total_probability']

# Derivation of general probability with formulation

print(f"\With the mentioned words, languages and sources, the total probability of fakeness is
{combined_probability:.2f}")
```

We finish everything by demonstrating the results of the robot script.

In this way, the most suspicious words in fake news were obtained: children, bribes, refugees, steel, classes, directors, Ukrainians, Poles, children, a large number.

The total probability of words being fake is 0.68.

Conclusions

A dataset analysis on "Fake news" was conducted based on language, source, and news text criteria. After that, the maximum probability that the news is fake based on specific words, languages, and sources was determined. Python programming language was also used to write the script.

A software product was obtained that allows you to identify words that appear in fake messages with a probability of 0.68 in current news.

Acknowledgements

The research was carried out with the grant support of the National Research Fund of Ukraine "Information system development for automatic detection of misinformation sources and inauthentic behaviour of chat users ", project registration number 187/0012 from 1/08/2024 (2023.04/0012).

References

1. ^*Дзялошинский И. Общественное мнение в ситуации тотальной дезинформации (проблема фейк-ньюз). International Scientific Journal of Media and Communications in Central Asia. 2024 Jun 11;(5). doi:10.62499/ijmcc.vi5.57.*
2. ^*Hobbs R. Teach the conspiracies. Knowledge Quest. 2017;46(1):16-24.*
3. ^*Kertysova K. Artificial intelligence and disinformation: How AI changes the way disinformation is produced disseminated, and can be countered. Security and Human Rights. 2018;29(1):55-81.*
4. ^*Nazarkevych M, Lytvyn V, Demchyk D. Ensemble Methods of Determining the Effective Activity of Enterprises. In: IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering; 2024 Feb; Cham: Springer Nature Switzerland. p. 160-183.*
5. ^*Nazarkevych M, Levush P, Pankovych B. Improving the Efficiency of Information Collection Based on the Development of a Chatbot with a Parser. In: 2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT); 2021 May; IEEE. p. 125-129.*