The object of this study is a disinformation detection process based on search algorithms for identifying fake news. The main task was to define a set of criteria and parameters for detecting the Ukrainian-language disinformation based on machine learning. A methodology has been considered for developing and filling a dataset of fakes for further training of the model and testing it for the purpose of identifying disinformation and propaganda, as well as determining the attributes of primary sources and routes of their distribution. This makes it possible to reasonably approach the definition of a model for forecasting the development of information threats in the cyberspace of Ukraine. In particular, the accuracy of automatic detection of the probability of disinformation in texts can be increased. For the English-language texts using balanced datasets for training when applying classical machine learning classifiers, the accuracy of identification and recognition of fakes is ≥90 %, and for the Ukrainian-language texts – ≥52 % and ≤90 %. That has made it possible to devise requirements for the structure and content of a typical dataset of fakes in the period after the full-scale invasion of Ukraine. The practical result of this work is the designed decision-making support system for monitoring, detecting, recognizing, and forecasting information threats in the cyberspace of Ukraine based on NLP and machine learning. The implementation of preliminary processing of the Ukrainian-language news, taking into account the linguistic features of the language in the text, increases the accuracy of fake identification by ≈1.72 times. Approaches to the construction of models for forecasting the development of information threats in cyberspace have been developed, which is an urgent task when fake news and information manipulation can affect public sentiment, politics, and the economy

Keywords: information threat, fake news, machine learning, disinformation detection, dataset, cyber security

UDC 004.89
DOI: 10.15587/1729-4061.2024.317456

DEVISING A METHOD FOR DETECTING INFORMATION THREATS IN THE UKRAINIAN CYBER SPACE BASED ON MACHINE LEARNING

Victoria Vysotska
PhD, Associate Professor*
Mariia Nazarkevych
Corresponding author
Doctor of Technical Sciences, Professor*
E-mail: mariia.a.nazarkevych@lpnu.ua
Serhii Vladov
PhD
Department of Scientific Activity Organization
Kremenchuk Flight College of Kharkiv National University of Internal Affairs
Peremohy str., 17/6, Kremenchuk, Ukraine, 39605
Olga Lozynska
PhD, Associate Professor*
Oksana Markiv
PhD, Associate Professor*
Roman Romanchuk
Head of Projects and Programs in the Field of Non-Material Production
LLC TIETO UKRAINE SUPPORT SERVICES
Heroiv UPA str., 72, Lviv, Ukraine, 79018
Vitalii Danylyk
Full-Stack Developer
Genesis Space
Olenivska str., 23, Kyiv, Ukraine, 04080
*Department of Information Systems and Networks
Lviv Polytechnic National University
S. Bandery str., 12, Lviv, Ukraine, 79013

Received 21.10.2024
Received in revised form 13.11.2024
Accepted 26.11.2024
Published 03.12.2024

***How to Cite**: Vysotska, V., Nazarkevych, M., Vladov, S., Lozynska, O., Markiv, O., Romanchuk, R., Danylyk, V. (2024). Devising a method for detecting information threats in the ukrainian cyber space based on machine learning. Eastern-European Journal of Enterprise Technologies, 6 (2 (132)), 36–48. https://doi.org/10.15587/1729-4061.2024.317456*

1. Introduction

Given the intensive development of information propagation in open access, the issue of security of distributed information systems, decision support systems (DSSs), and web resources is extremely relevant [1]. In particular, the aspect of cyber threats, which are modified and developed on an ongoing basis, is timely. Based on the information arrays, the following types of cyber attacks, which are most often observed in the web space, can be distinguished, namely phishing attacks, cyber terrorism, cyber espionage, and cyber crimes [2]. All the above types of cyber threats mainly concern [3]:

– blocking the work of state bodies;
– organization of failures in the provision of electronic services;
– improper use of e-mail;
– breach of confidentiality and integrity of data;
– informational pressure on society;
– psychological pressure on society;
– encroachment on the national information space of the state;

Copyright © 2024, Authors. This is an open access article under the Creative Commons CC BY license

– destruction or partial blocking of the work of strategically important state enterprises;

– destruction or partial blocking of high-risk facilities or life support systems.

Cyber attacks are a global problem today, and cyberspace is actively used by terrorist organizations for illegal purposes and unauthorized use of web resources. Therefore, Resolution of the Cabinet of Ministers of Ukraine No. 299 dated 04.04.23 [4] is an important stage in the fight against cyber threats and an important element of cyber defense. It allows resistance to the encroachment of intruders on the integrity of the state, especially under the conditions of a full-scale invasion of the territory of Ukraine. The application of artificial intelligence technologies in the issue of cyber attacks and the use of information space to strengthen the conduct of military operations obviously contribute to the deepening of the problem. It must be solved by combating distributed attacks on operational networks and supply chains using cloud services [1]. In particular, it is necessary to focus on countering encryption viruses, universal malware, botnets, and production systems. In today's world, misinformation has become one of the key threats to society, as information from unreliable sources spreads quickly through the Internet and social networks [5]. False or distorted information can have serious consequences for public opinion, politics, economy, and security [6]. A variety of actors – from private individuals to government structures – can deliberately spread disinformation to manipulate or create chaos. One of the most difficult aspects of countering disinformation is its detection and separation from reliable facts [7]. Under conditions when the volume of information is constantly growing, the automation of processes for verifying the veracity of texts is an urgent need [1–3]. Detection of disinformation by automating the text analysis process is relevant for a wide range of users, in particular [8]:

– journalists – for quick fact-checking and reducing the risk of disseminating inaccurate information;

– researchers – to analyze texts and identify potential sources of misinformation in academic or social studies;

– media analysts – to monitor the media space and identify trends in the spread of disinformation;

– ordinary users – to independently check the information they consume through social networks or news portals.

Therefore, research into the detection of information threats in the global cyberspace is relevant and promising given the conditions of modern digital era. Such research is especially relevant in the modern realities of information warfare, when fake news and manipulation of facts affect public opinion, public sentiment, politics, and the economy.

2. Literature review and problem statement

Study [9] defines propaganda based on analysis of political news in 35,993 articles based on TF-IDF and logistic regression (31,972/4,201 non-propaganda/propaganda articles, respectively). The accuracy for the training data is 0.9433254618697041. But for test data – 0.7332361516. The model successfully categorized 1917/205 respectively non-propaganda/propaganda articles (2012 total). But 585/146 respectively propagandistic/non-propagandistic articles are incorrectly categorized (731 in total). The accuracies for the training data in identifying Slavic and Germanic languages are similar (0.9899 and 0.9433) based on TF-IDF and logistic regression. And for the English test texts, the accuracy is almost twice as high as for Slavic texts (0.4856 and 0.7332).

In study [10], the Ukrainian-language/Russian-language fake news is recognized. The focus of the work is the features of NLP processing of textual content in the Slavic language for identification of fakes and propaganda based on the modified Porter stemmer, Word2vec library, TfidfVectorizer, and TF-IDF. The logistic regression method was used (accuracy for test data is 0.4856230031948816, and for training data – 0.9899319488817891). The almost twofold decrease in accuracy is explained by the complexity of processing and analyzing the Ukrainian-language text, the presence of informational noise, and synomization.

Paper [11] recognizes fake news for the year 2021 about COVID-19 based on TF-IDF+SVM (accuracy 84.29 %) and CNN+LSTM (accuracy 92.3 %). The older the data, the more accurate it is, so the more accurate the result. It is more difficult to conduct research on new data during an information war, when it is difficult to determine what is fake and what is reliable information.

In study [12] based on classical learning methods (multilayer perceptron, Bayes classifier, random forest, logistic regression, KNN, and decision tree), the obtained accuracy outperformed that reported in work [11] based on deep learning methods. All resulting accuracies range from 0.913 to 0.999. In [13], fake news recognition accuracy of 94–96 % was obtained based on CNN with 3 epochs and LSTM with 4 epochs (GloVe and TF-IDF). Work [14] demonstrates the analysis of news based on methods such as Multidimensional Scaling (MDS) and SVM classifiers with an accuracy of more than 90 % of fake identification. The resulting accuracy is not sufficient for such tasks. Conclusion: if you do not choose the best NLP methods for pre-processing data sets and/or do not form/select a complete/accurate data set, the accuracy of fake detection will not improve significantly. So to speak, there are errors in the input data (incomplete or inaccurate data sets) and models of the previous NLP method of news processing. This affects the accuracy of training models and the subsequent detection of fake/non-fake news. Emphasis should be placed on the application of the sentiment analysis method after preliminary processing of texts to form input data for classifiers.

It is necessary to improve the technology of detecting misinformation in the Ukrainian-language fake news based on refining the method of pre-processing the text and choosing the best method for training the system. Each paper [9–18] shows different accuracies of detecting fakes in the English-language news based on training on different data sets, as well as on different NLP and machine learning methods. The problem is the lack of prompt formation of such Ukrainian-language data sets for training systems, as well as their timely and periodic renewal in a short period of time, that is, the implementation of the best data collection pipeline. The second problem is the selection of the best NLP methods, in addition to the classical methods for pre-processing the Ukrainian-language texts, noise removal, duplication, and markup. It is necessary to generate dictionaries of noun groups, and not just alphabetic-frequency dictionaries of words. Implementation of the processes related to lemmatization, tokenization, separation of proper names, stemming, etc. is much more complicated for the Ukrainian-language texts. A significant advantage for such research is to perform semantic or ontological analysis to identify relationships between entities and distribute weights between

them. Theoretically, the method of sentiment analysis should give better accuracy in detecting a fake because propaganda and fakes are usually written with a negative coloration.

3. The aim and objectives of the study

The purpose of our work is to devise a method for detecting misinformation in textual content based on NLP and Machine Learning, taking into account sources, distribution channels, and stylistic features. This will make it possible to monitor, identify, and forecast information threats in Ukraine's cyberspace in real time based on machine learning.

To achieve the goal, the following tasks were set:

– to devise general functional requirements for the architecture of DSS for the detection of disinformation as an information threat in the form of false or propaganda content in Internet cyberspace;

– to build models for identifying informational threats;

– to devise a method for determining the probability of disinformation and preliminary processing of news to analyze the presence of fake content;

– to develop software for monitoring and detecting information threats in cyberspace;

– to analyze the results of experimental verification of the proposed method for detecting information threats based on the constructed fake news dataset.

4. The study materials and methods

The object of our research is the processes of monitoring, detecting, and forecasting information threats in the global cyberspace.

The main hypothesis of the study assumes that the use of an improved method for preliminary processing of the Ukrainian-language textual content could increase the accuracy of disinformation detection. The accuracy result can also be affected by the selection of the training model, the population of the dataset to train the model, and the balancing of this dataset before training the model. The selection of the best models, methods, and tools of textual content analysis methods and the training of a misinformation detection model should significantly improve the results of the accuracy of automating the decision-making process based on the constructed dataset.

The main methods and means for monitoring and detecting misinformation on the Internet are Natural Language Processing (NLP), machine learning, fact-checking, monitoring of user behavior, blockchain, and crowdsourcing platforms for fact-checking. Text analysis using artificial intelligence (AI) and NLP is used to look for attributes of manipulation or fake information. For example, AI can detect suspicious speech patterns that are typical of disinformation. Fact-checking, as a fact-checking based on DSS, occurs when information is quickly compared with reliable sources and determined if it is reliable. For this, databases (DBs) with verified information and algorithms for its analysis are used. In social networks, on the basis of a huge amount of statistical data, it is possible to monitor the behavior of users, for example, the activity of users, identifying the spreaders of disinformation, and detecting networks engaged in the manipulation of mass consciousness. Blockchain as a linked list has its own hash-sum to ensure information transparency. Blockchain can be used to confirm the authenticity of information sources, which reduces the number of fake news, since all information will be transparently tracked. There are also crowdsourced platforms for fact-checking where users can verify information themselves and provide their results, also effectively helping to identify disinformation. At the same time, neural networks are usually used to detect disinformation in textual content, as well as in images and videos, based on the processing and analysis of big data. Learning on the basis of labeled data through neural networks is trained on a large number of examples of true and false information. During training, the model analyzes various characteristics of the text – vocabulary, syntax, presentation style, as well as sources of information. The model then learns to distinguish between true and false information based on these features. Classification algorithms such as the support vector method, decision trees, and deep neural networks are used to build models that categorize texts as true/false based on statistical features. It is advisable to use natural language processing (NLP) based on the analysis of linguistic features, including the analysis of text content for emotional color, level of bias, degree of confidence or uncertainty in the presentation of facts, to identify information threats [19]. For example, fake information often contains sensational or emotionally charged headlines and phrases. The detection of keywords and phrases based on NLP technology helps find patterns or keywords that are often used in fake news [20], including elements of conspiracy theories or exaggerated claims. Checking links and sources based on machine learning can automatically not only find links to information sources but also check their authenticity using the database of reliable news outlets or official sources [21].

Algorithms for comparing information with other sources and available facts to identify contradictions are useful for checking news that is distributed in social networks [22]. Search models use metadata (time of creation/publication/rewrite, geographic location, history of changes) to identify suspicious materials [23]. In particular, news from new/unknown accounts is flagged as potentially fake. Some systems use artificial intelligence (AI) algorithms to analyze the writing style and identify possible attributes of automatic content generation or the use of bots to analyze and identify content authorship [24]. Analysis of images and videos is based on methods for recognizing changes in content, such as deepfake. They are used to analyze connections between users and news [25]. Fake information is often spread through botnets or fraudulent accounts, and graph neural networks can help detect these anomalies. Big Data analysis is based on the processing of large volumes of data to identify trends and anomalies that may indicate misinformation [26]. Tamper detection technologies are used to detect fake videos or images, such as deepfakes. AI analyzes changes in image pixels, inconsistencies in shadows, facial movements, etc.

Effective development of disinformation detection methods requires a combination of technological innovations with international cooperation, regulatory measures, and increasing the level of digital literacy of users [27]. The technologies also take into account the context in which the news is shared and social factors to better assess the likelihood that the information is fake [28]. Fakes are constantly evolving, so AI models need to be regularly updated based on new data to keep classification relevant. These technologies help significantly improve the ability to detect fake information and ensure the veracity of content on the Internet (Fig. 1).

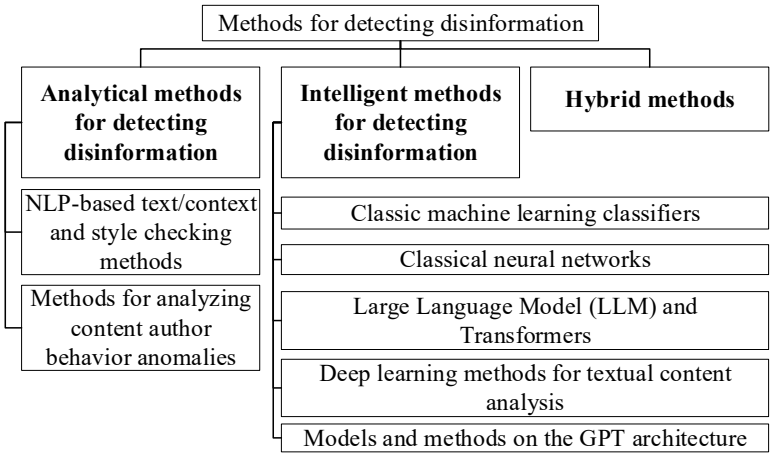


Fig. 1. Basic methods for identifying disinformation and information threats

When categorizing information into true and false, AI uses such linguistic aspects as vocabulary, syntax, and presentation style [29]. These elements help the algorithms determine whether the information meets the attributes of truth or contains typical patterns for misinformation. They use a set of parameters and criteria both for detecting disinformation and for sources of generating fakes and ways of their distribution. They mainly analyze words/phrases/topics/slang [30] to which there is a systematic or anomalous reaction of the majority of users (accounts) of the social community. This, in turn, leads to activation of abnormal inauthentic behavior of a group of social network participants, etc. An additional parameter is a change in the dynamics of activity depending on the time of day/week or holidays/weekdays, as well as a change in the frequency of posts/reposts per day. This helps identify the set of content that is created, posted, and re-posted by a directed group of social community participants with inauthentic behavior, including bots (programs).

Lexical analysis singles out a set of words that are made by a group of fake generators or embedded in bot programs, and also makes it possible to identify a collection of key phrases and patterns from the so-called "methodology" of the propaganda community. Fake news often uses emotionally charged or sensationalist language to gain attention. Algorithms detect such words as "shock", "sensation", "unknown", "conspiracy". Such terms often signal audience manipulation. Information that repeats established conspiracy or manipulative phrases may also be flagged as suspicious.

When analyzing syntax, attention is paid to the level of complexity and length of formed sentences, their construction. Fake information often has a simpler sentence structure to be understood and easily understood by a wide audience. Detecting too simple or too complex syntax can be an indicator of manipulation. Misinformation often contains inconsistent statements or overly vague wording. This may be a signal that the information is distorted or false. Analysis of the presentation style of the text content (article, message, post) makes it possible to single out the indicators and criteria of the emotional coloring of the content, the level of the language of persuasion and the presence of the so-called stylistic techniques of the author/bot.

Fake news often has an emotionally charged presentation style that forces audiences to respond to information intuitively rather than critically. Algorithms can determine the level of emotional load of a text by analyzing factors such as the use of evoked emotions (fear, anger, joy). Texts containing manipulative elements usually actively use persuasive expressions, such as "everyone knows", "without a doubt", "obviously". That makes it possible to hint at the undeniable veracity of information without providing evidence. Algorithms can also determine whether a text is an attempt at manipulation through rhetorical devices such as repetition or rhetorical questions, which are often used to influence emotions or create a desired impression. Fake news can use an overly dramatic style, simplistic syntax, and sensationalist language to spread quickly on social media. Such news is aimed at evoking emotions and often contains exaggerated claims. True news, on the other hand, usually has a more restrained presentation style, more complex syntax, and neutral vocabulary. This allows the information to be understandable but not speculative or manipulative. AI analyzes these elements in order to automatically assess whether the text has attributes of true information or, on the contrary, contains features of fake news. Simple flaws in the generation of fake news are textual (language) markers of content (article, post, message, news, post, etc.), in particular, thematic narratives of a certain social group, for example:

– The Armed Forces of Ukraine in the Kursk region use Nazi symbols en masse;

– the site of temporary deployment of the Armed Forces of Ukraine in ... was hit;

– the goals of the special operation have been achieved;

– The Armed Forces of Ukraine mined several dams and bridges in ...;

– goals of SVO are de-Nazification and demilitarization;

– Kyiv in three days;

– in Lviv they eat babies, etc.

Additional criteria are phrases/set phrases/words as the author's assumption (anonymous source informs, it became known, verified/reliable anonymous source informs/reported, it is believed, it is said, they say, believe, most likely, etc.). Analysis of the details of the images that accompany the content of a potential fake could make it possible to be sure of its level of credibility, for example:

– signboards with inscriptions of local institutions and street names;

– road attributes and license plates to identify a part of the world and/or country/city;

– presence and proportions of shadows, as well as sharpness of objects/subjects in the photo.

Additional tools such as TinEye and Google Image Search will make it possible to find analogs of photos from which a forgery was made. Analysis of a potentially fake user account and its content is also an additional criterion for checking the authenticity of the information disseminated. Usually, a fake profile has incomplete or strange user data, low number of comments/likes/likes, limited access to posts, and no or too few followers and friends.

We shall use the Python Natural Language Toolkit (NLTK) package to build the corresponding applications. The use of neural networks and machine learning is based on labeled data. Neural networks are trained on a large number of examples of true and false information. During training, the model analyzes various characteristics of the

text – vocabulary, syntax, presentation style, as well as sources of information. The model then learns to distinguish between true and false information based on these features. Classification algorithms based on the method of support vectors, decision trees, and deep neural networks are used to build models that categorize texts as true or false based on statistical features. The text is characterized by a level of emotional coloring and can also indicate the degree of bias or truthfulness of the data, which allows us to draw conclusions about the fakeness of the source of distribution. The analysis of the linguistic features of the texts is precisely the main issue in the processing of natural language, which involves the following:

– identification and analysis of emotional parts of the text;

– search by keywords and phrases;

– the use of various techniques of neuro-linguistic programming, namely tokenization, stemming/lemmatization, morphological segmentation, morphological analysis, and analysis of sentiments in the text.

For the English language, the process of morphological analysis based on tokenization, lemmatization, and stemming is carried out without significant efforts and problems. For German, the English phrase "Natural Language Processing" would be "Verarbeitung natürlicher Sprache" (noun capitalization required). But in declension according to 4 cases, only adjectives change the ending in German. For example, only for "natürlich", where -er is the feminine genitive ending for the word "Sprache". Unlike Ukrainian, nouns in German do not change like in English. For a similar phrase in Polish "Przetwarzanie języka naturalnego", the words change endings depending on the case and gender of the noun. In French, the similar phrase "Traitement du langage naturel" has articles. In the Russian phrase "Processing natural language" all three words change depending on case and gender. Similarly, for the Ukrainian phrase "Processing natural language" in the nominative singular there are still 5 cases (according to the changes of words and their endings). There are also possible entries for a persistent key phrase, for example, "Natural language processing" or "Natural language processing" (for tokenization, it will be "Natural language processing"). Changing the endings in words and alternation of letters during declension significantly complicates the process of lemmatization and, accordingly, morphological analysis. That significantly complicates any further analysis of the test, including fake news for interpretation and accuracy in detecting disinformation. Without conducting a preliminary morphological analysis based on the modified Porter algorithm (Fig. 2), it is not possible for the Ukrainian-language texts to correctly tokenize and lemmatize, as well as to determine the set of keywords in messages and news. Often, when writing a fake in a non-native language, both grammatical and spelling mistakes are made. This is an additional parameter for determining the potential for a fake in the analyzed text content. One of the solutions is to develop an algorithm for correcting Ukrainian word errors based on the spelling dictionary. The dictionary is submitted as a separate text file and must contain more than 30,000 words of the Ukrainian language. The content of the dictionary is formed according to the existing dictionaries of the Ukrainian language. If morphological analysis is used to convert tokenized words into lemmas, it is advisable to have one dictionary both for lemmatization and for finding errors. There are many algorithms for searching and comparing data. The search task is reduced to the task of string analysis (selecting matching substrings of two strings). In the generalized problem, you need to localize all text comparison operations. There are different ways to solve this problem; basically, they come down to the tasks of text search and associative thinking.

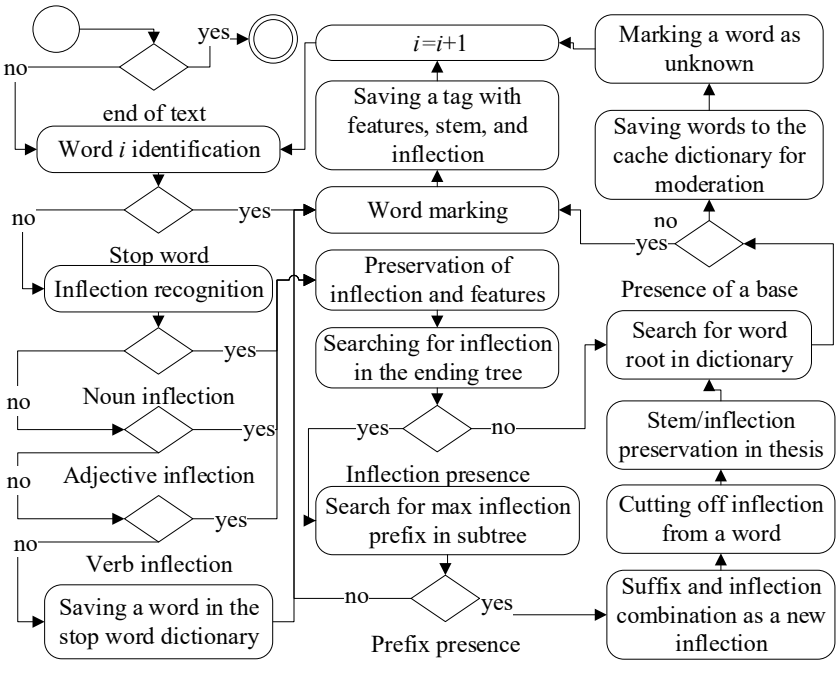


Fig. 2. Modified stemming algorithm for the Ukrainian-language texts

When designing a fake detection DSS with an error detection function, it is advisable to use the sorting algorithm due to its optimality. The time to find an error in one word lasts 0.1 seconds using a dictionary with 30,000 words. However, with the increase of words in the dictionary, the execution time of the algorithm will increase, although it will also give a reliable result. Therefore, when increasing the dictionary (up to 200,000 words), it is advisable to use the Knuth-Maurice-Prat algorithm. It runs faster and is considered the best for spell checking. Fact checking is also important for processing text information in open web resources. In particular, it is important to check the authenticity of sources, as well as to refute or confirm a fake. Usually, fact-checking resources are official or have a high level of reliability. It is important for high-fidelity processing and detection of contradictory parts of the text, manipulation, fakes, and propaganda, especially for checking posts in social networks. To this end, metadata analysis is widely used, namely establishing the time of publication; study of the history of changes; geographical location. AI for processing text information (Fig. 3) is also used for intelligent search, chatbots, content generation and analysis of the writing style of parts of the text.

There are the following 4 alternative options for DSS construction: A_1 – DSS of the combined type; A_2 – data-oriented; A_3 – knowledge-oriented; A_4 is model oriented. After the survey of experts, the matrices of pairwise comparisons were filled out, which made it possible to use MAI to analyze and choose the optimal option for DSS design (Table 1). The designed DSS is a web application that is cross-browser and multi-platform. The process of disinformation detection is implemented on the basis of NLP and machine learning methods. SQLite, Paraphrase Multilingual MiniLM L12 V2 (PMML12V2), FAISS, Scrapy and Tkinter were used to implement the key functions of DSS. SQLite is a lightweight relational database used to store text and metadata about information sources. SQLite makes it possible to store data locally, without the need to configure a separate database server. DSS uses SQLite to store analyzed texts, vector representations, and the reliability rating of each source. The PMML12V2 machine learning model is used to vectorize the input text.

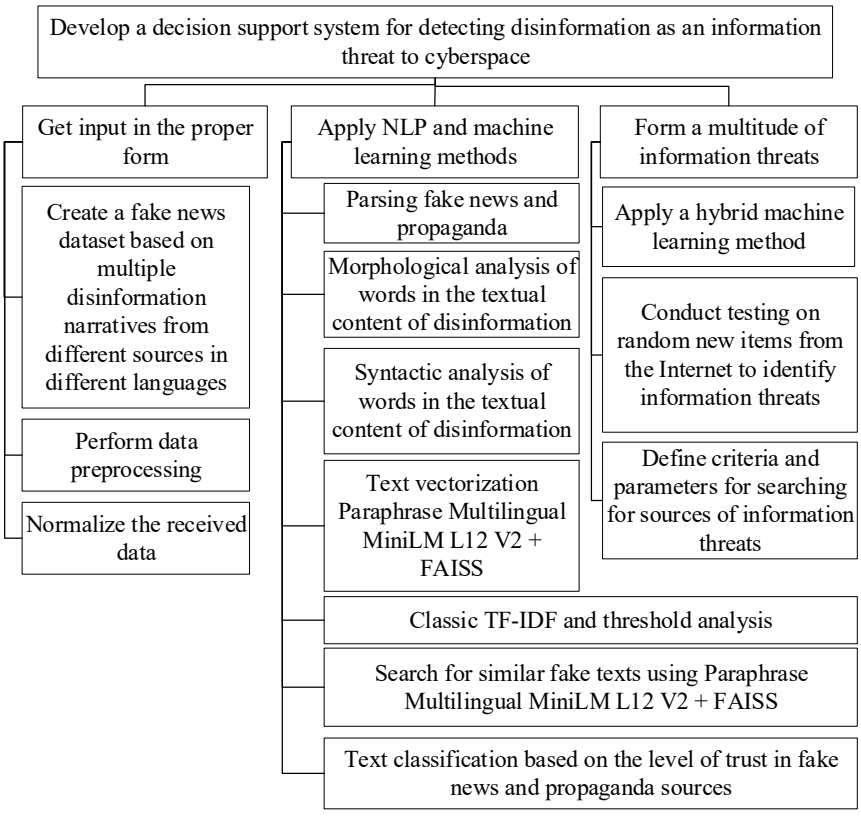


Fig. 3. A tree of goals for identifying cyberspace information threats

Table 1

Priority vector

Alternatives	A_1	A_2	A_3	A_4	Weight	Variant
A_1	1	2	6	2	0,42596	1
A_2	1/2	1	6	3	0,256486	2
A_3	1/6	1/6	1	2	0,252045	3
A_4	1/2	1/3	1/2	1	0,065509	4

The model makes it possible to transform text into multidimensional vectors that store semantic information. MiniLM L12 V2 was chosen due to its high accuracy when working with multilingual texts, including the Ukrainian language, which is important for the tasks set before the project. It is integrated using the transformers library. FAISS is for efficient search of similarity vectors in large datasets. After vectorizing the text with MiniLM, FAISS is used to quickly find texts that contain similar information. This tool makes it possible to significantly speed up the search process among a large number of records in the database. The Scrapy web parsing framework is used to collect textual information from various online sources. Scrapy automates the process of extracting content from websites, which makes it possible to constantly update the database of texts and sources assigned to different categories of trust. Tkinter provides a user interface of the program where the user can enter text to be analyzed, click the "Analyze" button, and view the results. Tkinter is an easy tool for building simple but effective GUIs, which makes interacting with DSS user-friendly.

5. Research results based on monitoring, detecting, and forecasting information threats in cyberspace

5. 1. General requirements for the architecture of a decision support system for disinformation detection

Existing combination of functions allows the program to detect texts that may be disinformation and provide the user with a clear assessment based on comparison with already known sources (Fig. 4). This makes the program a useful tool for analyzing and verifying information in real time. The DSS uses Scrapy to parse data and collect textual information from a variety of online sources that are pre-categorized by trust level. The configuration file defines a list of sources with a high trust rating (for example, news outlets or blogs with a proven reputation) and a low rating (resources that regularly spread misinformation). Data from these sources is parsed and stored in SQLite database along with information about the source and its rating. This process can run in the background, which makes it possible to constantly update the database of texts for analysis. The PMML12V2 model, which is part of the Sentence-BERT architecture, is used for text analysis and vectorization. This model works well with many languages, including Ukrainian, due to its multilingual nature. This model is able to capture the semantics of the text and create a multidimensional vector that describes the content of the text. Vectorization is a key step since it is vectors that are used to compare the entered text with the texts already in the database.

Conversion of the entered text is performed quickly and with high accuracy, providing adequate processing even for short or complex texts in the Ukrainian language. The FAISS tool is used to quickly and efficiently search for texts similar to the one entered by the user. It allows vector searches in large data sets with high performance. FAISS searches for

similar texts in the database using a vector representation of the entered text. As a result, DSS finds texts that have similar information and determines from which sources they were obtained. To categorize the text based on the level of trust in the sources, the found texts are mapped to the sources from which they were obtained, according to the information stored in the database. Each source has its own trust rating defined in the configuration file (for example, high trust sources have a rating of 1.0 and low trust sources have a rating of 0.0). DSS analyzes the number of found texts with different levels of confidence and based on this, determines the overall confidence in the entered text. DSS calculates the percentage of probability that the entered text is misinformation. This assessment is based on the ratio of found texts from reliable and unreliable sources. If most of the texts found come from sources with a high credibility rating, the probability that the text is true will be high. If most of the texts are similar to those coming from unreliable sources, the probability of misinformation increases. DSS displays the result in the form of a percentage, which allows the user to quickly assess the reliability of the entered text. In the process of designing the DSS, several approaches were tested to determine the probability of disinformation. The final version, which uses the PMML12V2 model and the FAISS similarity vector search tool, is the result of an evolution of approaches and improvements.

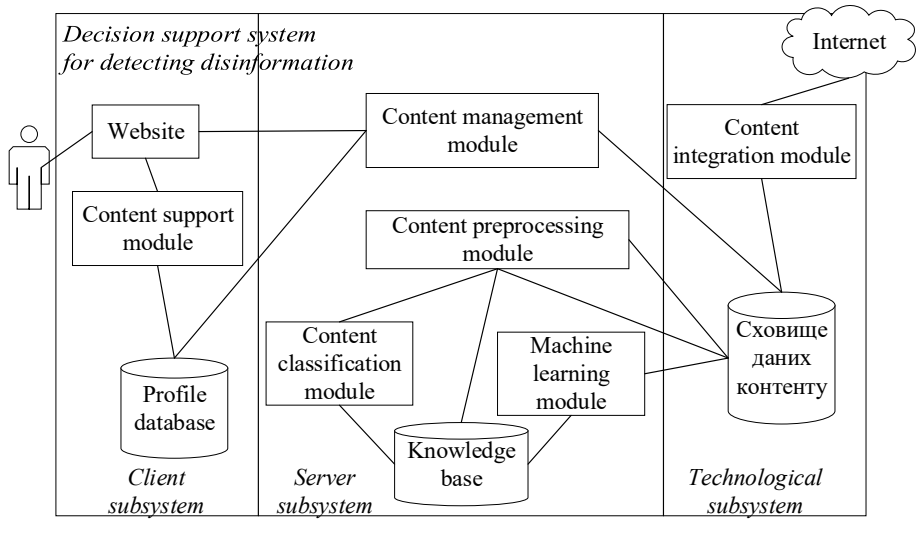


Fig. 4. General typical architecture of a decision support system for disinformation detection

5. 2. Models for determining informational threats

Model 1, based on TF-IDF (Term Frequency-Inverse Document Frequency) makes it possible to determine the importance of words in the text relative to the entire collection of documents. After calculating the TF-IDF, each text was compared by cosine similarity with the texts in the database. It makes it possible to determine the similarity between the entered text and texts from trusted/untrusted sources. The problem lies in weak understanding of semantics, lack of support for multilingualism, and low accuracy of identification of a Ukrainian-language fake.

Model 2, based on BERT (Bidirectional Encoder Representations from Transformers), allows better analysis of text semantics. The use of BERT made it possible to convert texts into vector representations, after which a linear search for the nearest vectors in the database took place. The problem is low search efficiency, high resource consumption, and insufficient support for the Ukrainian language. Linear vector search takes too much time when processing large text databases, which reduces the speed of analysis. Using the model is computationally expensive and requires significant resources to run. The model is not optimized for multilingual processing, it does not give the best results for the Ukrainian-language texts. This leads to a loss of accuracy when analyzing texts, especially in cases with complex linguistic structures and contexts.

Model 3, based on PMML12V2, provides high-precision multilingual vectorization of texts with significantly lower resource requirements. The model is optimized for working with texts in different languages, including Ukrainian, which makes it ideal for this application. Using MiniLM makes it possible to get accurate vector representations of texts for effective semantic analysis. To search for similar texts, FAISS was chosen, which significantly speeds up the search for vectors even in large data sets. The advantage of the approach is improved work with the Ukrainian-language texts, fast search, low resource requirements, and high accuracy. The PMML12V2 model provides accurate vectorization of the Ukrainian-language texts, which allows for better recognition of their content and semantics. Using FAISS to search for vectors makes it possible to significantly speed up the text analysis process, even with large data sets. MiniLM requires less computational resources compared to the BERT model, making it suitable for use on conventional machines without loss of accuracy. Optimized for multilingual texts, the MiniLM model significantly improved the accuracy of semantic text classification. The final approach, using MiniLM for vectorization and FAISS for searching, provides fast and accurate text analysis. An important advantage is the ability of the model to work effectively with the Ukrainian-language texts, which improves the accuracy of disinformation detection in Ukrainian cyberspace.

5. 3. Method for determining the probability of disinformation and preliminary processing of news to analyze the presence of a fake message

Stage 1. Data collection and processing takes place in the background, in parallel with the main actions of the user. DSS uses the Scrapy framework for automatic parsing of texts from web resources and defined in the configuration file. Sources are divided by trust level: reliable and unreliable (high/low rating). After the text is collected from the web resource, the process of vectorization using the PMML12V2 model immediately passes. This step transforms the text into a multidimensional vector that describes its semantic content. Vectorized texts together with metadata, including source identifiers, are stored in an SQLite database. It is important to note that the level of trust in the sources is not stored in the database but is determined during further analysis based on the information contained in the configuration file. The stage is implemented under a back-

ground mode, without interfering with the user's work with DSS. The stage formalizes the process of preparing a message from a chat or the Internet for the presence of fake information through a sequence of defined steps:

Step 1. 1. Uploading a message to the data processing DSS. A unique number is assigned to the downloaded message. The message is being checked for availability in the DSS, this will indicate that the message is a duplicate. Otherwise, the message is added to the message array for further processing. The function additionally stores counters to count duplicate statistics and the total number of messages. Assuming that the set of input data forming the set of messages K is as follows: $K=\{x_1, x_2, x_3, \dots, x_{i-1}, x_i\}$, where, for example x_i, are the words in the message.

Step 1. 2. Checking the dataset for duplicates is an important step in data preparation as it helps ensure the accuracy and reliability of the model. This is especially important for messages, as the presence of duplicates can negatively affect the model's learning process and its ability to generalize. The news dataset is described by the set $C=\{K\}$. Duplicates are defined as identical messages. The dependence $C=\{K_i | i=1, 2, ..., N\}$ was used to save the message in set C and check for duplicates. If the cardinality of set H (the number of unique elements) is less than N, it means that there are duplicates in the dataset. The similarity threshold τ was determined. If $\text{sim}(x_i) \geq \tau$, then x_i is considered a duplicate.

Stage 2. Balancing the data set of messages. The message dataset imbalance problem occurs when the number of messages in different classes is not the same. This can cause the learning model to bias the focus of the training model in favor of categories with more messages, which can reduce classification accuracy for less represented categories. Therefore, balancing the message dataset is an important step in data preparation for training machine learning models. Its essence is to equalize the number of words in different classes. This helps avoid biasing the model towards a particular class, which can occur due to imbalance in the data. Let there be a set of data in the dataset $K=\{x_1, x_2, x_3, ..., x_{i-1}, x_i\}$, where x_i is the news in the dataset. Then $D=\{(x_1, y_1), (x_2, y_2),..., (x_n, y_n),\}$ where y_i is the corresponding class label, such as fake/not fake or a specific news topic. If there are H_d classes, and each class d has λ_d news, then the data set balancing algorithm is as follows:

Step 2. 1. Determining the number of messages for each class λ_d:

$$\lambda_d : \lambda_D = \sum_{i=1}^{n} I(y_i = d), \quad (1)$$

where I is an indicator function equal to 1 if $y_i=d$ and 0 otherwise.

Step 2. 2. Determining the target number of messages N in each class. Let N be the average value of the desired number of messages in each class after balancing:

$$N = \frac{\sum_{j=1}^{H_d} \lambda_D}{H_D}. \quad (2)$$

Step 2. 3. Balancing of the dataset based on Oversampling (adding copies of existing messages from underrepresented classes) and Undersampling (removing some messages from overrepresented classes) methods. For each class d, if $n_d<N$, we add copies of existing messages (oversampling). If $\lambda_d>N$, we randomly select N messages (undersampling). That is, for each class d from $\lambda_d<N$, we add $N-\lambda_d$ copies of existing messages:

$$D' = D \cup \{(x_i, y_i) | y_i = d, copies = N - \lambda_d\}. \quad (3)$$

Stage 3. The user interacts with the program through a graphical interface, entering text for analysis. After entering the text, the DSS immediately starts the process of its vectorization. It uses the same PMML12V2 model that is used in the background to process texts from web resources. The model converts the entered text into a vector, which is then used to search for similar texts in the database. The user does not need to wait for the completion of the process of collecting and processing new texts since the database already contains enough vectorized texts for immediate search.

Stage 4. After vectorization of the entered text, the DSS uses the FAISS tool to search for similar vectors among those stored in the database. FAISS (Facebook AI Similarity Search) is designed for efficient and fast search for vectors among a large amount of data. The search algorithm determines the texts closest in terms of semantic content by comparing their vector representations. The result is a list of found texts together with metadata about the sources.

Stage 5. On the basis of the list of found texts, DSS performs the classification of the entered text. Each text found is associated with its source, which has a confidence level (high or low) defined in the configuration. The DSS determines the number of found texts from reliable and unreliable sources. The algorithm evaluates the percentage distribution of confidence levels among found texts similar in content:

Step 5. 1. If most of the found texts come from reliable sources (with a rating close to 1.0), the entered text is categorized as true.

Step 5. 2. If most of the texts found are associated with sources that have a low level of trust (a rating close to 0.0), the text is considered potentially disinformation.

5. 4. Software and main modules of the system for monitoring and detecting information threats in cyberspace

The DSS uses threshold values for classification, which are taken from the configuration file. For the true text, the share of reliable sources exceeds 80 %. For a fake, the share of unreliable sources exceeds 80 %. The result of the classification is displayed in the form of a percentage score, which shows the probability that the entered text is disinformation. This score helps the user quickly assess the reliability of the text. DSS is built on a modular architecture that provides a clear division of functionality between different components. Each module is responsible for performing certain tasks, and the main controller app.py coordinates their interaction (Fig. 5). The main app.py controller is responsible for coordinating all other modules. Main flow of execution:

– initialization of the graphic interface;

– calling functions for text collection, vectorization, and classification;

– obtaining results from modules and transferring them to subsequent components;

– management of background processes of data collection from web sources.

The graphic.py module is responsible for the graphics interface. It uses Tkinter to build a user interface through which text is entered for analysis; we start the text analysis process; view the result in the form of a percentage probability of a fake or error messages.

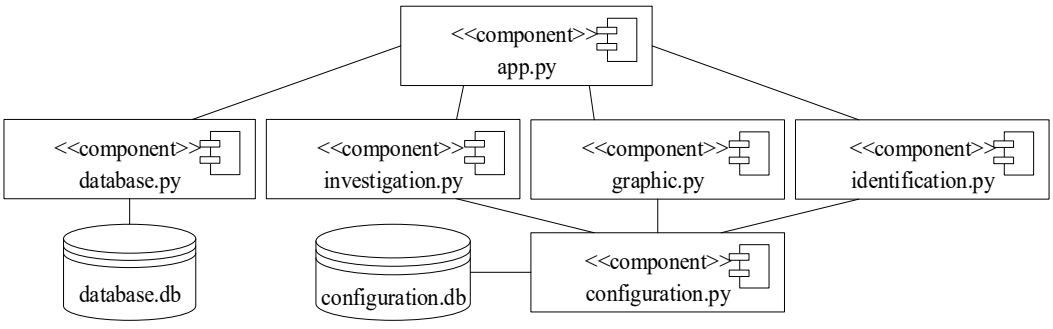


Fig. 5. Architecture of the disinformation detection system

The module for working with the SQLite database database.py is responsible for:

– building and maintaining a database for storing texts and their vector representations;

– recording collected and processed texts;

– data search in the database during the analysis of the entered text.

The key module for processing the entered text identification.py performs the following tasks as:

– text vectorization using the PMML12V2 model;

– searching for similar texts in the database using the FAISS tool;

– text classification based on the results found;

– calculating the probability of disinformation based on trust in the sources of the found texts.

The investigation.py module is responsible for collecting texts from web sources. It uses Scrapy to parse the sources specified in the configuration file. The collected texts are vectorized and stored in the database for further analysis. All processing takes place in the background, which does not interfere with the main process of analyzing the entered texts. The configuration.py module contains all configuration parameters of the application. This includes a list of sources for parsing and their confidence level; threshold values for categorizing texts as true or disinformation; operating parameters of FAISS and other instruments. When interacting between modules, app.py calls functions from graphic.py to interact with the user, and also receives input text for further processing. The app.py module passes the text to identification.py, where it is vectorized, then FAISS is used to search for similar texts in the database. After that, identification.py classifies the text and returns the results to app.py. The app.py module also interacts with investigation.py to receive new texts from web sources that are added to the database via database.py. The DSS processes two main streams of data (Fig. 6) as background collection and processing of texts, as well as analysis of the entered text. Web resources are parsed by the investigation.py module in the background, texts are vectorized and stored in the database for further use. The DSS immediately vectorizes it, searches for similar texts using FAISS, and performs classification based on the results found.

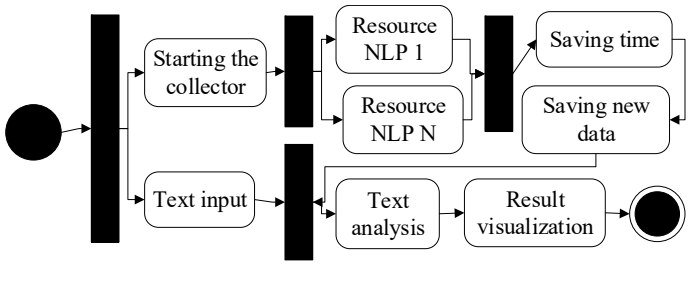


Fig. 6. Parallel processing flows of two main data streams

The results are returned to app.py, which passes them on for output through the GUI or for further processing by other modules.

5. 5. Requirements for the construction of a dataset of fake news and the process of training the model

At this stage of processing, it is necessary to conduct an analysis through additional subtasks of data balancing: data completeness analysis; removal of emissions; coding of categorical features; attribute scaling; data augmentation. To balance the data set, you need to add simulated messages to the less represented classes. The dataset contains 11 attributes with different message options. Also, the set contains 13,860 records with values for each news from February 2022 to October 2024 through the eyes of students, graduate students, and lecturers at Lviv Polytechnic University. The data was collected in the period September–October 2024. Competently and qualitatively constructed datasets of fakes, propaganda, and disinformation improve their use in DSS for the recognition and identification of fake news in the Internet space. According to the data, the main characteristics of the message include the parameters 'Number', 'Data', 'Time', 'Tekst', 'M', 'Pos1/Repos1', 'Author/Group', 'www', 'language', 'Like' , 'Post'. The data set has some imbalance as the value of fake news occurs 5980 times (43.1 %) and authentic news 7910 times (56.9 %). This factor must be taken into account during data pre-processing. 13,620 news items that were distributed in Ukrainian content from September 2024 to October 2024 were analyzed. The dataset was formed by teams of students, postgraduates, and lecturers at Lviv Polytechnic University. The dataset is constructed in such a way that information published as news on Internet sites (29 %), which distribute news, on social networks Telegram (41 %), Facebook (22 %), Twitter (6 %), Instagram (1 %), Vkontakte+Yandex (1 %). For the latter, such a small percentage is explained by the fact that on the territory of Ukraine, where the dataset was collected, access to these resources is officially blocked at the state level. The distribution in the dataset of true and fake messages from Facebook is 54 % and 46 %, respectively. The distribution of real and fake Telegram messages in the dataset is 46.9 % and 53.1 %, respectively. The distribution of true and fake messages from news outlets in the dataset is 23.2 % and 76.8 %, respectively. The distribution of messages in the dataset in the Ukrainian, Russian, and other languages is 79.8 %, 18 %, and 2.2 %, respectively. In this way, the dataset was formed on the grounds that there were equal numbers of "true" and fake messages. Among fake news, the words "warrior", "satisfied", "horror", "shock", "unbelievable", "rashishtka" were most often found.

When training a model, training data is formed at the first stage. Fake samples and their descriptions are needed to train the model. We divide the data into training and test sets – 70–80 % for training and 20–30 % for testing. We use a training set to train the model. In the next step, the data is cleaned and standardized for further use in the algorithms. The message undergoes linguistic processing to highlight key characteristics. By tokenization, we divide the text into separate words or phrases. The text is converted into numerical vectors using the TF-IDF, Word2Vec, or PMML12V2 methods. Next, the model is trained on those data with corresponding fake/not fake labels. The fake/not fake label for new messages is assigned after the model is trained. We use the model to predict the presence of a fake in messages for which there are no labels. To assign a label, existing categorized messages with known labels are used to train the classification model. The model learns to categorize messages based on their descriptions and other characteristics. Using historical data, classification algorithms are trained on input textual and numerical characteristics of messages. The models analyze the relationship between the textual description and the corresponding label. When a new message (news) arrives in DSS, the algorithm automatically generates a proposal for fake or not fake as a label based on the description and other characteristics. If the DSS is unsure of the result, it can provide several label options for further selection by the user. The results of automatic classification are checked by experts. Algorithms can self-learn from new cases, improving accuracy over time (Fig. 7–10). The more news descriptions with known labels used for training, the better the model will get for new data:

1. For models 1–2, when predicting the English-language fakes in the initial stages, the results of the accuracy value are 0.52 (Fig. 7) for a dataset of 8980 records (4668 fakes and 4312 not fakes). After training the first model and testing it on new data and supplementing the dataset, the following results were obtained for various samples from the dataset Accuracy – 0.791, and for the second model Accuracy – 0.998.

2. For models 1–3, when predicting the Ukrainian-language fakes at the initial stages, the results of the accuracy value are 0.52 (Fig. 8) for a dataset of 13,620 records (5,980 fakes and 7,910 non-fakes). After training the first model and testing it on new data and supplementing the dataset, we got the following results for different samples from the dataset Accuracy – 0.52, for the second model Accuracy – 0.846, and for the third – 0.895.

Manual testing of model 3 was conducted to identify the Ukrainian-language fake news from the data set that was not used during training for two different disinformation texts (Fig. 9). Testing was also conducted on current news. During this testing, it was found that any text unknown to the model is identified as fake news.

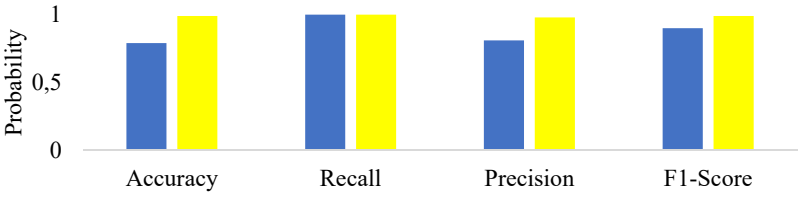


Fig. 7. Results of identifying the English-language fake news for models 1–2

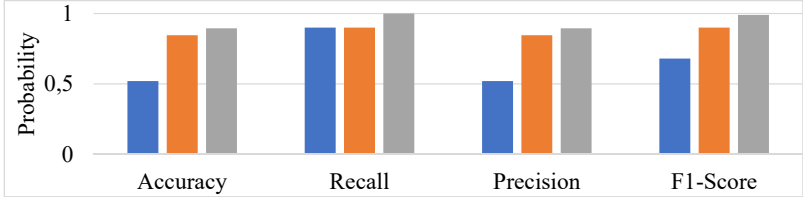
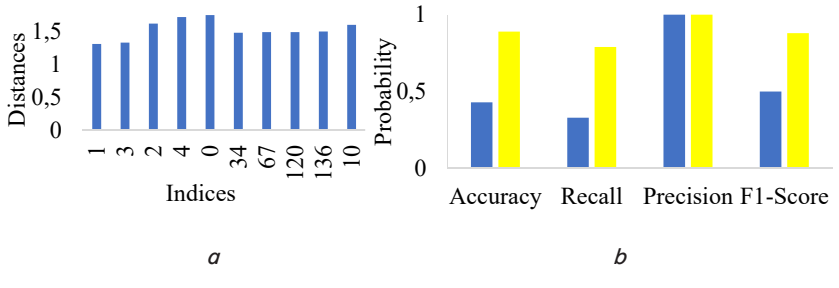


Fig. 8. Results of identifying the Ukrainian fake news for models 1–3

Fig. 9. Results of manual classification of the Ukrainian news for models 1–3

In the following experiments, models were investigated on the Ukrainian-language news in real time (Fig. 10). After training the second model based on Porter's modified algorithm (Fig. 2) and testing it on new data and adding to the dataset, Accuracy 0.846 was obtained for different samples from the dataset. The Precision values for the true/false class are 0.78/1.00, respectively. Recall values are 1.00/0.67 and F1–Score are 0.88/0.80, respectively. The nearest neighbor indices are [32, 33, 12, 29, 30], and the nearest neighbor distances are [1.2567494, 1.4715298, 1.4769558, 1.5156906, 1.5193214].



a *b*

Fig. 10. Plot: *a* – distances to nearest neighbors; *b* – evaluation metric of models 2–3 during manual testing

After training the third model and testing it on new data and supplementing the dataset based on balancing the message dataset according to formulas (1) to (3), we got Accuracy 0.895 for different samples from the dataset. The Precision values for the true/false class are 0.83/1.00, respectively. Recall values are 1.00/0.79 and F1–Score are 0.90/0.88, respectively. Nearest neighbor indices are now [146, 46, 58, 79, 56], and all nearest neighbor distances tend to 0.

6. Discussion of results based on monitoring, detecting, and forecasting information threats in cyberspace

For the English-language texts using balanced datasets for training when applying classic machine learning classifiers, the accuracy of identification and recognition of fakes is often ≥90 % [9–14]. For example, for the forecast model based on the BOW function and Logistic Regression, the results of accuracy value are 0.98. For detecting a fake, the precision value is 0.98, recall 0.99, and F1–score 0.98 with support 2971. For detecting a non-fake, the precision value is 0.99, recall 0.98, and F1–score 0.98 with support 3029. This is explained by the fact that Porter's classic algorithm for the English texts is simple, does not require the presence of large dictionaries of all possible word bases and of all possible endings, as in the Ukrainian language. In total, for the morphological analysis of the Ukrainian nouns, about ≈1,300 rules for processing suffixes and endings are used, taking into account the alternation of letters. There are about ≈100 rules of morphological analysis for adjectives, and ≈800 for verbs. That's why you can't just use the BOW bag of words to process the Ukrainian-language fake news. Similarly, this applies to inflections in the Ukrainian words b (number of different words ≈290), ш (≈120), ц (≈40), ф (≈220), ж (≈360), г (≈640), л (≈760), з (≈590), с (≈920), п (≈350), ч (≈960), н (≈2540), р (≈2700) and д (≈1050). If we take only the letter я (≈164062) as the ending of a Ukrainian word, then for other variations the ending will be ся (≈148160), ося (≈30770), мося (≈20540), ймося (≈6840), мемося (≈1640), имемося (≈1620), тимемося (≈1620), атимемося (≈1240), ватимемося (≈800), уватимемося (≈560), etc. The principle is to parse all words from back to front to the maximum possible ending according to the tree of all inflections in the Ukrainian language, selecting the base of the word and checking it with the dictionary of all word bases. There are no such complications in the English language.

Our Ukrainian-language fake detection method is implemented on the basis of the modified Portrait algorithm, dataset balancing, and the PMML12V2 machine learning model, which is used for vectorization of the entered text. The model works well with many languages, including Ukrainian, due to its multilingual nature. This model is able to capture the semantics of the text and create a multidimensional vector that describes the content of the text. Vectorization is a key step since it is vectors that are used to compare the entered text with the texts already in the database. Conversion of the entered text is performed quickly and with high accuracy, providing adequate processing even for short or complex texts in the Ukrainian language. The key module for processing the entered text identification.py performs the following tasks:

– text vectorization using the PMML12V2 model;

– searching for similar texts in the database using the FAISS tool;

– text classification based on the results found;

– calculating the probability of disinformation based on trust in the sources of the found texts.

On the basis of investigating known and our own dataset of fakes, DSS for monitoring and detecting information threats in cyberspace has been designed for the automatic detection of the probability of disinformation in texts. The DSS combines modern NLP technologies, machine learning algorithms, and effective search using FAISS, which makes it possible to quickly analyze texts in the Ukrainian language. The use of the multilingual MiniLM model makes it possible to correctly work with texts in several languages, which expands the scope of its application. After training the first model and testing it on new data and supplementing the dataset, the following results were obtained for various samples from the dataset: Accuracy – 0.52; for the second model, Accuracy – 0.846, and for the third – 0.895.

The implementation of pre-processing of news based on the modified Porter algorithm, taking into account linguistic features of the language of a text before learning the Ukrainian-language fake identification model 3, increases the accuracy of fake identification by ≈1.72 times compared to model 1. Compared to the use of classical lemmatization and stemming methods, the accuracy of fake identification increases for the Ukrainian-language texts, but not as much as for the English-language texts. For the English-language information, compared to the Ukrainian-language information, the accuracy of detecting a fake is greater by ≈0.2 according to works [9–14].

The main limitations of our study are an insufficient set of data (less than 5,000 authentic/fake news, confirmed by experts), complexity, and low resources for studying the Ukrainian language. Also, a limitation inherent in this study is the insufficient number of conducted experiments on various models and classifiers for improving the technology of disinformation detection. Most of the research focuses on studying the English-language news, not Ukrainian-language, although one of the centers of information warfare is the cyberspace of Ukraine.

The disadvantage is that the dataset is not collected by experts, such as journalists, psychologists, sociologists, or political scientists, but by simple average readers of different age groups from the same geographical space. With better and more accurate data in voluminous datasets, the results of information threat detection accuracy could be better. Due to the fact that the dataset was not collected by experts, but by ordinary citizens of Ukraine from one region, its accuracy is influenced by the opinion of a limited circle of people. In simple words, what is a fake for person *X* (for example, a student), for person *Y* (for example, a lecturer) is not a fake. Therefore, there are possible errors in the dataset for marking similar news with different labels as fake or not fake. This in turn affects the accuracy of results when training the model. Although even with this drawback, the advantage of our study is the improved accuracy of disinformation detection for the Ukrainian-language content.

Prospects for further research include experimental testing with increased datasets (more than 5,000 records) and neural network training on at least 5 news message topics. Also, the dataset should be collected by representatives of different geographical locations and different professions. The problem of such studies is still the lack of general requirements and rules for building and filling such datasets. In turn, to eliminate this problem, it is necessary to conduct many experiments with smaller datasets to find their advantages and disadvantages. This would reveal patterns, parameters, and criteria for improving the method of identifying disinformation and information threats, sources of distribution, and inauthentic behavior of chat users.

7. Conclusions

1. On the basis of our analysis of existing strategies for the anti-disinformation plan and the peculiarities of the

types of fakes, the general architecture of DSS for the detection of disinformation as an information threat has been designed. This has made it possible to define a set of criteria and parameters for disinformation detection. A methodology for building and filling a dataset of fakes for further training of the model and its testing for the purpose of identifying disinformation and propaganda, as well as determining the attributes of primary sources and routes of their distribution, has been considered.

2. Based on the review of existing methods of intellectual search for disinformation and the features of fact-checking sites, models of forecasting the development of information threats in the cyberspace of Ukraine were studied. For the English-language texts using balanced datasets for training when applying classic machine learning classifiers, the accuracy of identification and recognition of fakes is ≥90 %, and for the Ukrainian-language texts – ≥52 % and ≥90 %. This made it possible to devise requirements for the structure and content of a typical dataset of fakes in the period after the full-scale invasion of Ukraine.

3. On the basis of the proposed models for detecting information threats, a method for determining the probability of disinformation and preliminary processing of news to analyze the presence of a fake message has been devised. Balancing the news dataset is an important step to ensure an even representation of all classes. This helps create a more robust and fairer model that is unbiased towards certain classes due to data imbalances. The implementation of preliminary news processing, taking into account the linguistic features of the language of the text, increases the accuracy of fake identification by ≈1.72 times.

4. Based on the study of known and our own dataset of fakes, DSS for monitoring and detection of informational threats in the cyberspace of DSS was designed for the automatic detection of the probability of disinformation in texts. The DSS combines modern NLP technologies, machine learning algorithms, and effective search using FAISS, which makes it possible to quickly analyze texts in the Ukrainian language. The use of the multilingual MiniLM model makes it possible to correctly work with texts in several languages, which expands the scope of its application. After training the first model and testing it on new data and supplementing the dataset, we got the following results for different samples from the dataset: Accuracy – 0.52, for the second model, Accuracy – 0.846, and for the third – 0.895.

5. Experiments were conducted on the constructed dataset using machine learning algorithms. For the English-language information, compared to the Ukrainian-language information, the accuracy of fake detection is greater by ≈0.2. To detect an English-language fake, the value of accuracy is 0.98, precision is 0.98, recall is 0.99, and F1-score is 0.98. To detect a Ukrainian-language fake, the value of accuracy is 0.846, precision is 1.00, recall is 0.67, and F1-score is 0.80.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The research was carried out using the grant support from the National Research Fund of Ukraine, project registration number 187/0012, dated 1/08/2024 (2023.04/0012) "Development of an information system for automatic detection of sources of disinformation and inauthentic behavior of chat users" under the competition "Science for strengthening the defense capability of Ukraine" .

Data availability

The data will be provided upon reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Acknowledgments

This paper was prepared using the grant support from the National Research Fund of Ukraine, project registration number 187/0012, dated 1/08/2024 (2023.04/0012) "Development of an information system for automatic detection of sources of disinformation and inauthentic behavior of chat users" under the competition "Science for Strengthening Ukraine's Defense Capability". We would also like to thank the reviewers and editors for their precise and concise recommendations that improved the representation of results.

References

1. Trofymenko, O. H. (2019). Monitorynh stanu kiberbezpeky v Ukraini. Pravove zhyttia suchasnoi Ukrainy. Mizhnar. nauk.-prakt. konf. Vol. 1. Odesa: VD «Helvetyka», 642–646. Available at: https://dspace.onua.edu.ua/items/3aa8c85a-0013-4a36-9c74-bedbcd915593
2. Trofymenko, O., Prokop, Y., Loginova, N., Zadereyko, O. (2019). Cybersecurity of Ukraine: analysis of the current situation. Ukrainian Information Security Research Journal, 21 (3). https://doi.org/10.18372/2410-7840.21.13951
3. Yashchuk, V. I. (2024). Rol ta mistse stratehiyi kiberbezpeky ukrainy u zabezpechenni informatsiynoi bezpeky derzhavy. Available at: https://sci.ldubgd.edu.ua/jspui/bitstream/123456789/13824/1/1%20Yashchuk_monogr_rozdil13.pdf
4. Deiaki pytannia reahuvannia subiektamy zabezpechennia kiberbezpeky na rizni vydy podiy u kiberprostori (2023). Postanova Kabinetu Ministriv Ukrainy vid 04.04.23 r. No. 299. Available at: https://zakon.rada.gov.ua/laws/show/299-2023-п#Text
5. Vysotska, V., Chyrun, L., Chyrun, S., Holets, I. (2024). Information technology for identifying disinformation sources and inauthentic chat users' behaviours based on machine learning. CEUR Workshop Proceedings, 3723, 427–465. Available at: https://ceur-ws.org/Vol-3723/paper24.pdf

6. Vysotska, V., Przystupa, K., Chyrun, L., Vladov, S., Ushenko, Y., Uhryn, D., Hu, Z. (2024). Disinformation, Fakes and Propaganda Identifying Methods in Online Messages Based on NLP and Machine Learning Methods. International Journal of Computer Network and Information Security, 16 (5), 57–85. https://doi.org/10.5815/ijcnis.2024.05.06
7. Khairova, N., Galassi, A., Lo, F., Ivasiuk, B., Redozub, I. (2024). Unsupervised approach for misinformation detection in Russia-Ukraine war news. Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems. Volume IV: Computational Linguistics Workshop. https://doi.org/10.31110/colins/2024-4/003
8. Wierzbicki, A., Shupta, A., Barmak, O. (2024). Synthesis of model features for fake news detection using large language models. Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems. Volume IV: Computational Linguistics Workshop. https://doi.org/10.31110/colins/2024-4/005
9. Oliinyk, V.-A., Vysotska, V., Burov, Ye., Mykich, K., Basto-Fernandes, V. (2020). Propaganda Detection in Text Data Based on NLP and Machine Learning. CEUR workshop proceedings, 2631, 132–144. Available at: https://ceur-ws.org/Vol-2631/paper10.pdf
10. Vysotska, V., Mazepa, S., Chyrun, L., Brodyak, O., Shakleina, I., Schuchmann, V. (2022). NLP Tool for Extracting Relevant Information from Criminal Reports or Fakes/Propaganda Content. 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 93–98. https://doi.org/10.1109/csit56902.2022.10000563
11. Dar, R. A., Hashmy, Dr. R. (2023). A Survey on COVID-19 related Fake News Detection using Machine Learning Models. CEUR Workshop Proceedings, 3426, 36–46. Available at: https://ceur-ws.org/Vol-3426/paper4.pdf
12. Mykytiuk, A., Vysotska, V., Markiv, O., Chyrun, L., Pelekh, Y. (2023). Technology of Fake News Recognition Based on Machine Learning Methods. CEUR Workshop Proceedings, 3387, 311–330. Available at: https://ceur-ws.org/Vol-3387/paper24.pdf
13. Afanasieva, I., Golian, N., Golian, V., Khovrat, A., Onyshchenko, K. (2023). Application of Neural Networks to Identify of Fake News. CEUR Workshop Proceedings, 3396, 346–358. Available at: https://ceur-ws.org/Vol-3396/paper28.pdf
14. Shupta, A., Barmak, O., Wierzbicki, A., Skrypnyk, T. (2023). An Adaptive Approach to Detecting Fake News Based on Generalized Text Features. CEUR Workshop Proceedings, 3387, 300–310. Available at: https://ceur-ws.org/Vol-3387/paper23.pdf
15. Saquete, E., Tomás, D., Moreda, P., Martínez-Barco, P., Palomar, M. (2020). Fighting post-truth using natural language processing: A review and open challenges. Expert Systems with Applications, 141, 112943. https://doi.org/10.1016/j.eswa.2019.112943
16. Elzayady, H., Mohamed, M. S., Badran, K. M., Salama, G. I. (2022). Detecting Arabic textual threats in social media using artificial intelligence: An overview. Indonesian Journal of Electrical Engineering and Computer Science, 25 (3), 1712–1722. http://doi.org/10.11591/ijeecs.v25.i3.pp1712-1722
17. Shahbazi, Z., Byun, Y.-C. (2021). Fake Media Detection Based on Natural Language Processing and Blockchain Approaches. IEEE Access, 9, 128442–128453. https://doi.org/10.1109/access.2021.3112607
18. Guo, Z., Schlichtkrull, M., Vlachos, A. (2022). A Survey on Automated Fact-Checking. Transactions of the Association for Computational Linguistics, 10, 178–206. https://doi.org/10.1162/tacl_a_00454
19. Liu, X., Qi, L., Wang, L., Metzger, M. J. (2023). Checking the Fact-Checkers: The Role of Source Type, Perceived Credibility, and Individual Differences in Fact-Checking Effectiveness. Communication Research. https://doi.org/10.1177/00936502231206419
20. Martín, A., Huertas-Tato, J., Huertas-García, Á., Villar-Rodríguez, G., Camacho, D. (2022). FacTeR-Check: Semi-automated fact-checking through semantic similarity and natural language inference. Knowledge-Based Systems, 251, 109265. https://doi.org/10.1016/j.knosys.2022.109265
21. Ali, F., El-Sappagh, S., Islam, S. M. R., Ali, A., Attique, M., Imran, M., Kwak, K.-S. (2021). An intelligent healthcare monitoring framework using wearable sensors and social networking data. Future Generation Computer Systems, 114, 23–43. https://doi.org/10.1016/j.future.2020.07.047
22. Camacho, D., Panizo-LLedot, Á., Bello-Orgaz, G., Gonzalez-Pardo, A., Cambria, E. (2020). The four dimensions of social network analysis: An overview of research methods, applications, and software tools. Information Fusion, 63, 88–120. https://doi.org/10.1016/j.inffus.2020.05.009
23. Daud, N. N., Ab Hamid, S. H., Saadoon, M., Sahran, F., Anuar, N. B. (2020). Applications of link prediction in social networks: A review. Journal of Network and Computer Applications, 166, 102716. https://doi.org/10.1016/j.jnca.2020.102716
24. Chen, Q., Srivastava, G., Parizi, R. M., Aloqaily, M., Ridhawi, I. A. (2020). An incentive-aware blockchain-based solution for internet of fake media things. Information Processing & Management, 57 (6), 102370. https://doi.org/10.1016/j.ipm.2020.102370
25. Avelino, M., Rocha, A. A. de A. (2022). BlockProof: A Framework for Verifying Authenticity and Integrity of Web Content. Sensors, 22 (3), 1165. https://doi.org/10.3390/s22031165
26. Wang, X., Xie, H., Ji, S., Liu, L., Huang, D. (2023). Blockchain-based fake news traceability and verification mechanism. Heliyon, 9 (7), e17084. https://doi.org/10.1016/j.heliyon.2023.e17084
27. Boyen, X., Herath, U., McKague, M., Stebila, D. (2021). Associative Blockchain for Decentralized PKI Transparency. Cryptography, 5 (2), 14. https://doi.org/10.3390/cryptography5020014
28. Xue, J., Wang, Y., Tian, Y., Li, Y., Shi, L., Wei, L. (2021). Detecting fake news by exploring the consistency of multimodal data. Information Processing & Management, 58 (5), 102610. https://doi.org/10.1016/j.ipm.2021.102610
29. Sahoo, S. R., Gupta, B. B. (2021). Multiple features based approach for automatic fake news detection on social networks using deep learning. Applied Soft Computing, 100, 106983. https://doi.org/10.1016/j.asoc.2020.106983
30. Kaliyar, R. K., Goswami, A., Narang, P., Sinha, S. (2020). FNDNet – A deep convolutional neural network for fake news detection. Cognitive Systems Research, 61, 32–44. https://doi.org/10.1016/j.cogsys.2019.12.005