UDC 004.85
https://doi.org/10.31891/csit-2023-1-6

Oleksandr IERMOLAIEV, Inessa KULAKOVSKA
Petro Mohyla Black Sea National University

# IMPROVING THE QUALITY OF SPAM DETECTION OF COMMENTS USING SENTIMENT ANALYSIS WITH MACHINE LEARNING

*Nowadays, people spend more and more time on the Internet and visit various sites. Many of these sites have comments to help people make decisions. For example, many visitors of an online store check a product's reviews before buying, or video hosting users check at comments before watching a video. However, not all comments are equally useful. There are a lot of spam comments that do not carry any useful information. The number of spam comments increased especially strongly during a full-scale invasion, when the enemy with the help of bots tries to sow panic and spam the Internet. Very often such comments have different emotional tone than ordinary ones, so it makes sense to use tonality analysis to detect spam comments. The aim of the study is to improve the quality of spam search by doing sentiment analysis (determining the tonality) of comments using machine learning. As a result, an LSTM neural network and a dataset were selected. Three metrics for evaluating the quality of a neural network were described. The original dataset was analyzed and split into training, validation, and test datasets. The neural network was trained on the Google Colab platform using GPUs. As a result, the neural network was able to evaluate the tonality of the comment on a scale from 1 to 5, where the higher the score, the more emotionally positive the text and vice versa. After training, the neural network achieved an accuracy of 76.3% on the test dataset, and the RMSE (root mean squared error) was 0.6478, so the error is by less than one class. With using Naive Bayes classifier without tonality analysis, the accuracy reached 88.3%, while with the text tonality parameter, the accuracy increased to 93.1%. With using Random Forest algorithm without tonality analysis, the accuracy reached 90.8%, while with the text tonality parameter, the accuracy increased to 95.7%. As a result, adding the tonality parameter increased the accuracy for both models. The value of the increase in accuracy is 4.8% for the Naive Bayes classifier and 4.9% for the Random Forest.*

*Keywords: sentiment analysis, spam detection, neural network, text analyze, Python.*

Олександр ЄРМОЛАЄВ, Інеса КУЛАКОВСЬКА
Чорноморський національний університет імені Петра Могили

# ПОКРАЩЕННЯ ЯКОСТІ ПОШУКУ СПАМУ В КОМЕНТАРЯХ ЗА ДОМОГОГОЮ АНАЛІЗУ ТОНАЛЬНОСТІ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

*У наш час люди все більше і більше проводять часу в Інтернеті та відвідують різноманітні сайти. Багато з цих сайтів мають коментарі, що допомагають людям приймати рішення. Так, багато відвідувачів інтернет-магазину дивиться на відгуки до товару перед покупкою, а користувачі відеохостингів часто орієнтуються на коментарі перед переглядом. Проте не всі коментарі однаково корисні, досить часто можна зустріти спам-коментарі які не несуть жодної корисної інформації. Особливо сильно зросла кількість спам-коментарі під час повномасштабного вторгнення, коли ворог за допомогою ботів намагається посіяти паніку та заспамити Інтернет простір. Часто такі коментарі відрізняються за емоційним забарвленням від звичайних, тому існує сенс використовувати аналіз тональності для їх виявлення. Метою дослідження є покращення якості пошуку спаму за допомогою визначення тональності коментарів з використанням машинного навчання. В результаті було обрано LSTM нейромережу та датасет для її навчання та перевірки. Було описано три метрики для оцінки якості нейромережі, а датасет було проаналізовано та розбито та навчальну, валідаційну та тестову вибірки. Навчання нейромережі відбувалося на платформу Google Colab з використанням GPU. У результаті нейромережа змогла оцінювати тональність коментаря по шкалі від 1 до 5, де чим вище оцінка – тим більш емоційно-позитивний відгук і навпаки. Після навчання нейромережа досягла точності у 76.3% на тестовому датасеті, а серденя квадратична помилка становила 0.6478, що позначає що нейромережа помиляється менше ніж на один клас. При використанні алгоритму наївного байєсівського класифікатора без аналізу тональності, точність склала 88.3%, тоді як з параметром тональності тексту точність зросла до 93.1%. При використанні алгоритму випадкового лісу без аналізу тональності, точність склала 90.8%, тоді як з параметром тональності тексту точність зросла до 95.7%. В результаті що додавання параметру тональності підвищило точність для обох моделей. Значення приросту точності становить 4.8% для наївного байєсівського класифікатора та 4.9% для випадкового лісу.*

*Ключові слова: аналіз тональності, пошук спаму, нейромережі, аналіз тексту, Python.*

## Introduction

The main feature of modern AI algorithms is that they can "accumulate experience". In this way, they are able to solve some tasks with informal conditions, which is not able to do any productive, but rigidly programmed computer system.

A neural network consists of a system of connected and interacting simple processors called neurons. They are usually quite simple, especially compared to the processors used in personal computers. Each neuron of such a network connects only with signals that it receives and signals that it sends to other neurons [1]. Nevertheless, being connected in a rather large network, such individual simple parts together are capable of performing quite complex tasks.

The fields of application of neural networks are quite diverse – these are text and speech recognition, semantic search, expert systems and decision support systems, stock price prediction, security systems, text analysis, etc. This study considers an example of using a neural network to analyze the tone of comments.

Probably, there are tasks for neural network in each subject area. Here is a list of individual areas where the solution of this kind of tasks is of practical importance already now: economy and business, medicine, communication

and the Internet, production automation, political and sociological technologies, security and security systems, input and processing of information, geological exploration.

Classification tasks are understood as tasks for dividing a set of input signals into a predetermined number of classes [2]. After training, such a network is able to determine to which class the input signal belongs. In some varieties, the neural network can signal that the input signal does not belong to any of the selected classes – this is a sign of the appearance of new data that is not in the training sample, or of incorrect input data.

The field of text tonality analysis is quite new, and new technologies appear in it every year. Currently, one of the most popular tools for this is Cognitive Service from Microsoft Azure (Fig. 1). However, it has several disadvantages. One of them is that it is part of the Microsoft Azure cloud platform, so in order to use it, you need to familiarize yourself with the basics of this platform and configure access. In addition, the neural networks for this service were trained on the texts of articles from the Internet, so this service will have less accuracy on reviews. Instead, the field of spam detection is popular and there are many proven solutions on the market. One of them uses Google Gmail to filter spam in emails. However, it is not known whether it use tonality analysis for spam detection.

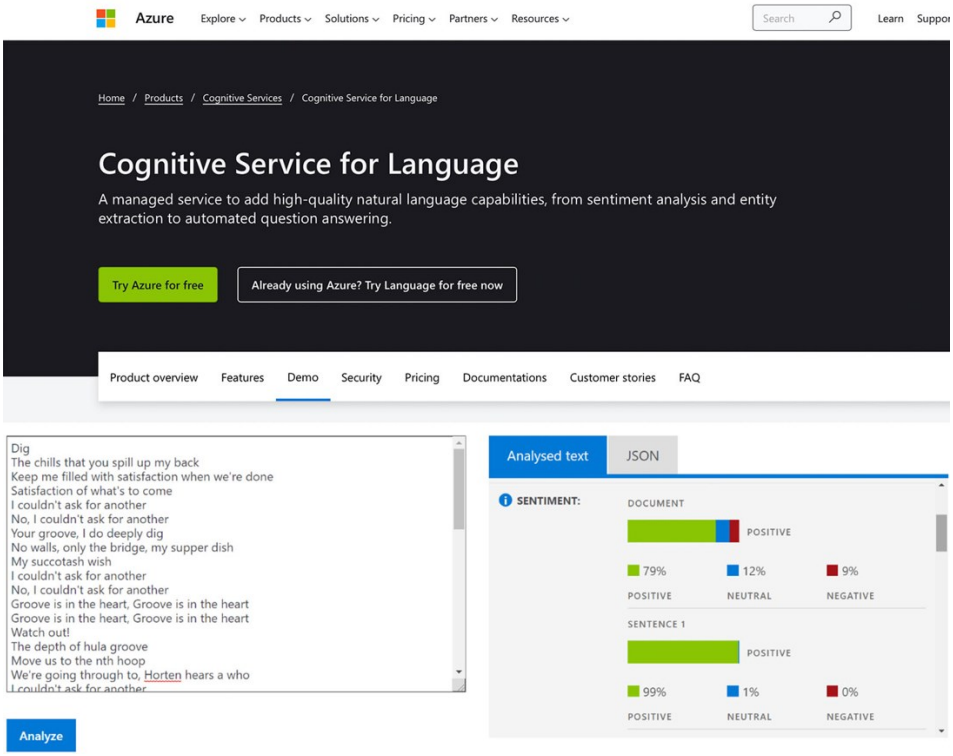


**Fig.1. Similar system – Microsoft Azure Cognitive Service**

One of the tasks that artificial intelligence can solve is the analysis of the tonality of the text (sentiment analysis) [3]. This means analyzing the emotional coloring of the text, taking into account its context. Tonality is the emotional attitude of the author of the statement towards some object expressed in the text. The emotional component expressed at the level of a lexeme or communicative fragment is called lexical tonality (or lexical sentiment). The tonality of the entire text as a whole can be determined as a function (in the simplest case, as a sum) of the lexical tonalities of its constituent units (sentences) and the rules for their combination. The main goal of tonality analysis is to find thoughts in the text and identify their properties. In this research, the tonality analysis was used to assign the text to one of five classes. So, it turns the task into classification problem.

### Data preparation and neural network creation

We decided to use recurrent neural network, especially a structure called LSTM. This choice was justified by the ability of these neural networks to remember the last analyzed words [4], and therefore to remember the context.

Firstly, it was necessary to choose a dataset. The right dataset is a very important part of training a neural network. It must follow next conditions:

– have a sufficient amount of data;
– these data must be up-to-date;
– the dataset must be balanced, have approximately the same amount of data in each class;
– must be collected from a reliable source.

In addition, the dataset needs to be divided into several parts. Train sample is a sample for which training takes place. Each training iteration will process this dataset, or a certain part of it. This part of the dataset should be

the largest. It is very important that it meets the requirements described above, as this part of the dataset will be responsible for the accuracy of the model.

When model is built according to the training sample, then the quality assessment of this model, made on the same sample, turns out to be more optimistically [5, 6]. This phenomenon is called overtraining and in practice it occurs very often. A good empirical assessment of the quality of the constructed model is provided by its verification on independent data that were not used for training. Therefore, as described in it, in order to avoid the phenomenon of overtraining, it is necessary to check the neural network on another part of the data, which is called validation. This part of the dataset is significantly smaller than the training sample.

A test or control dataset is a sample that evaluates the quality of the constructed model. If the training and validation samples are fed to the model input multiple times, it is very important that the network sees the test dataset as few times as possible. It is on the basis of this sample that the final accuracy of the neural network will be evaluated.

To solve the task of comment analysis, we chose a fairly well-known dataset [7], which meets all the criteria described above. The dataset itself consists of 700,000 reviews that were taken from the famous USA site yelp.com. In addition to the text, each comment has a tonality rating from 1 to 5.

Preprocessing and training took place in Jupyter Notebook. It is an interactive computing environment that allows users to write, run, and share code, and display and manipulate data in an interactive and collaborative manner. Jupyter Notebook supports a variety of programming languages, including Python. One of the key strengths of Jupyter Notebook is its ability to interactively calculate each step. In addition, it has powerful built-in data visualization capabilities.

The input data has been cleaned and tokenized. In cleaning stage, we removed all non-alphanumeric characters from each part of the text. Tokenization is the process of breaking text into smaller pieces called tokens [8]. In this case, the tokens will be individual words. Therefore, further each comment is divided into separate words. We use an English dictionary from nltk library to do tokenization. It is used to bring all the words of the main form. This process is called stemming and is an important part of text normalization [9]. Since there is no ideal algorithm for finding the basic form, we use English dictionary from nltk that containing words and their various forms. This stemming method is called table lookup. The advantages of this method are the simplicity, speed and convenience of handling exceptions for each language. The disadvantages include the fact that the search table must contain all forms of words, which means that the algorithm will not work with new words.

After that, we convert each word into a vector using "word2vec" algorithm. It accepts a large text corpus as input data and assigns a vector to each word, giving the coordinates of the words at the output. For this task, the dictionary contained 2000 words that were most frequently encountered among all comments. All fewer common words were replaced with the key "UNK". During text analysis, large datasets can contain thousands or even millions of unique words, many of which are irrelevant or do not add significant information to the analysis. When building a model, it is important to keep the number of parameters to a minimum, as each new parameter complicates the model and increases the risk of overtraining. Conversely, by limiting the number of parameters, it is easier and faster to train the model. It can also reduce the computational cost of training the model, as well as reduce the memory required to store the feature matrix. This is especially important when training a neural network on a GPU in Google Colab, as there are limits on the maximum time of use of computing resources.

We created a network accepts a vector of a comment text and has five outputs. Each one matches rating from 1 to 5. The output with the highest score is the most likely class number. After several training iterations, a dropout method was added. This is a regularization method used in neural networks to prevent overtraining. It works by randomly removing a certain percentage of neural network connections during training. This forces the network to create redundant correct connections, making it more robust and less likely to overtrain. The dropout value is a hyperparameter and is determined by experiment [3]. A value of 0.2 was chosen for this neural network.

The entire learning process was divided into epochs. In each epoch, the training dataset is divided into groups of 2000 comments. Since the training sample contains 80% of 650,000 comments, then the total count of groups is $\frac{0.8 \cdot 650000}{2000} = 260$. Validation occurs at the end of each epoch. There are three metrics used in validation: the logistic loss function, the precision and the root mean square error (RMSE). Training and verification of the neural network took place on the Google Colab cloud platform. The result of studying is shown on Figure 2.

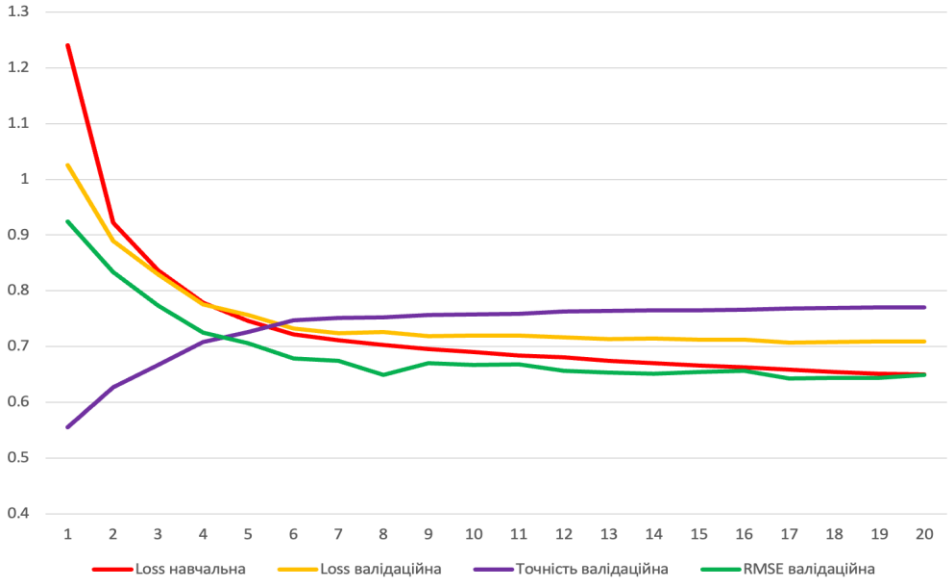


**Fig.2. Progress of neural network training**

As a result, the accuracy was 76.3%. The value of the logistic loss function metric and the root mean square error were 0.7061 and 0.6478, respectively.

## Experiments and researching

The next part of the work is about using the created neural network for improving accuracy of spam detection. A well-known dataset [10] with comments from the youtube.com website was chosen. The dataset itself consists of 1961 reviews, each of which belongs to the spam or non-spam (ham) class. Both classes are represented in the dataset in the same proportion. Just like the previous dataset, this one has also been normalized and vectorized [8, 9]. In addition, all Internet references were selected from the text. To perform tokenization, the TfidfVectorizer function from the sklearn library is used. It analyzes the number of repetitions of each word and leaves only a set of the most popular words. Next, this algorithm assigns a specific vector to each word. The resulting vector representations of words can be used in natural language processing and machine learning. In this case, the simplest way of converting words into a vector was used, when each word is assigned its own vector, which does not depend on other words.

Firstly, the Naive Bayes classifier was first used. A Naive Bayes classifier is a probabilistic classifier based on the Bayes theorem, that based on naive assumptions of independence [11]. The MultinomialNB object from the sklearn library was used for training. As an input, it accepted a vector in which the text is encoded, and as an output it outputs the probability of this text belonging to spam, from 0 to 1. In the result, the accuracy of this method is 88.3%.

In the next step, we add a feature of the sentiment analyze to learning parameters. This feature was calculated using a previously created neural network. As a result, accuracy increased from 88% to 93%. Confusion matrices before and after are shown on Figure 3.

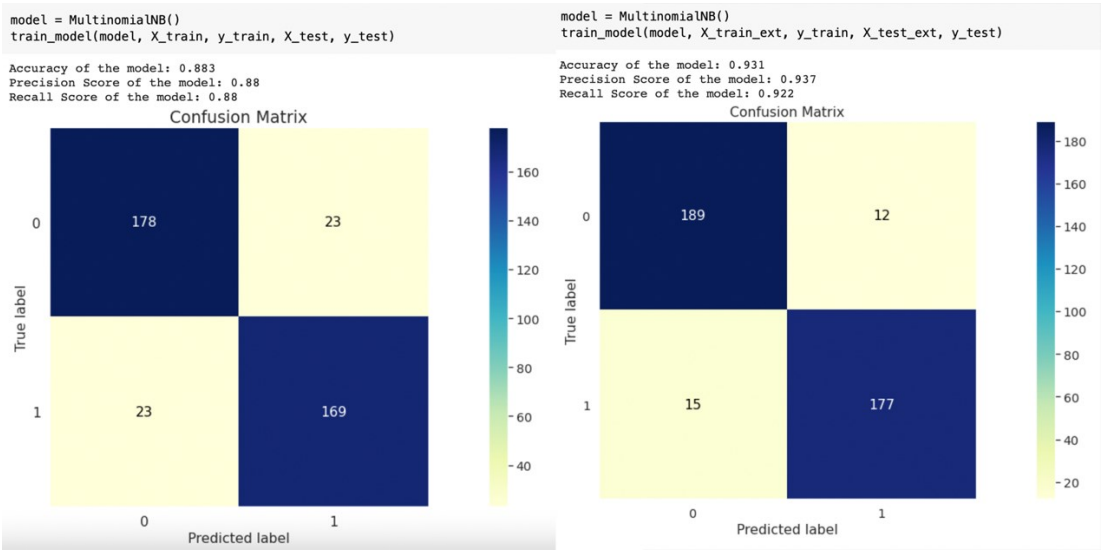


**Fig.3. Confusion matrices for Naive Bayes before and after addition a feature of sentiment analysis**

The same flow was done for the Random Forest algorithm. Random Forest is a supervised machine learning algorithm used for classification and regression tasks [1]]. It is an ensemble learning method that combines multiple decision trees to make predictions. In a Random Forest model, a set of decision trees is built independently, each using a random subset of the features and a random subset of the training data. Each decision tree is trained to predict the outcome variable based on the subset of features and data it was given. As a result, the accuracy increased from 91% to almost 96%. Confusion matrices before and after are shown on Figure 4.

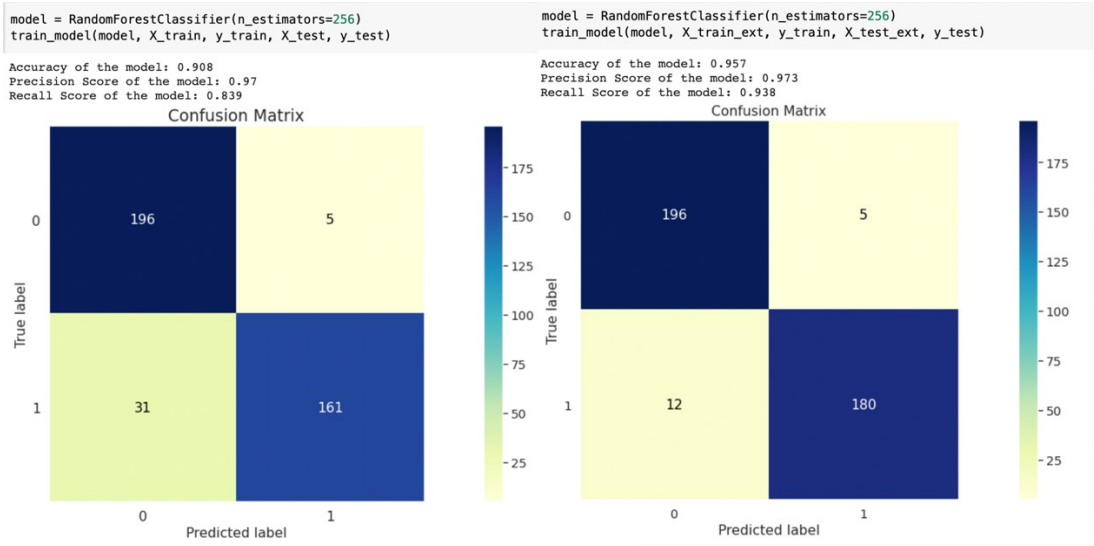


**Fig.4. Confusion matrices for Random Forest before and after addition a feature of sentiment analysis**

## Conclusions

Firstly, we made an analysis of the subject area of systems for analyzing the tone of the text. It showed that there is no universal solution. The opposite situation is for the task of spam detection, for which there are many ready-made systems, but they have a close (private) program code. Therefore, it is not known whether they use the analysis of the tonality of the text or not.

Secondly, we used an LSTM neural network and dataset [7] for its training and validation. Three metrics for evaluating the quality of a neural network were described, and the dataset was analyzed and split into training, validation, and text datasets. The neural network was trained on the Google Colab platform using GPUs. As a result, the neural network was able to evaluate the tone of the comment on a scale from 1 to 5, where the higher the rating, the more emotionally positive the response and vice versa. After training, the neural network achieved an accuracy of 76.3% on the test dataset, and the root mean squared error was 0.6478. This means that error of neural network is less than one class.

Thirdly, we researched how this neural network can improve accuracy of spam detection. In order to do this, a dataset [10] was selected. The text of each response of this dataset was cleaned, tokenized and transformed into a vector in the same way as the previous dataset. When we trained the Naive Bayes classifier algorithm without sentiment analysis, the accuracy was 88.3%, while with the text sentiment analysis the accuracy increased to 93.1%. When we trained the Random Forest without sentiment analysis, the accuracy was 90.8%, while with the text sentiment analysis the accuracy increased to 95.7%.

As a conclusion, the adding feature of sentiment analysis increases the accuracy for both models. The value of the increase in accuracy is 4.8% for the Naive Bayes classifier and 4.9% for the Random Forest. Therefore, sentiment analysis can be used to improve spam detection. It is worth noting that the accuracy of the Random Forest is higher than the accuracy of the Naive Bayes classifier for this task.

## References

1. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities, 1984. С. 147-169.
2. Segaran T. Programming collective intelligence. LA, 2012
3. Deerwester, S. C., Dumais S. T., Landauer T. K. Indexing by Latent Semantic Analysis. 1990. С. 391–407.
4. Gers F. A., Schmidhuber J., Cummins F. Learning to forget: continual prediction with LSTM. UK, 1999. С. 850–855.
5. Goodfellow I., Bengio Y., Courville A. Deep Learning, 2016. 773 с.
6. John E. Kelly I. Steve Hamm Smart Machine, 2014. 147 с.
7. Yelp review full dataset: веб-сайт. URL: http://hidra.lbd.dcc.ufmg.br/datasets/yelp_2015/original/yelp_review_full_csv.tar.gz (дата звернення 01.02.2023).
8. Yang Y., Pedersen J. O. A comparative study of feature selection in text categorization. 1997. С. 412–420.
9. Yang Y., Pedersen J. O. Feature selection in statistical learning of text categorization. – 1997.
10. YouTube Spam Collection Data Set: веб-сайт. URL: https://archive.ics.uci.edu/ml/datasets/Youtube+Spam+Collection (дата звернення 01.02.2023).
11. Alzubi, J., Nayyar, A., Kumar, A. Machine learning from theory to algorithms: an overview, 2018. 43 с.

| | | |
|---|---|---|
| **Oleksandr Iermolaiev**<br>**Олександр Єрмолаєв** | Master of Computer Science<br>e-mail: abionics.dev@gmail.com<br>https://orcid.org/0009-0008-1092-2101 | Чорноморський національний університет імені Петра Могили |
| **Inessa Kulakovska**<br>**Інеса Кулаковська** | PhD of Physical and Mathematical Sciences,<br>Department of Intelligent Information Systems, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine<br>e-mail: Inessa.Kulakovska@chmnu.edu.ua<br>https://orcid.org/0000-0002-8432-1850 | Чорноморський національний університет імені Петра Могили |