



The Method dynamic TF-IDF

Oleg Barabash¹, OleksandrLaptiev², Oksana Kovtun³,
Olga Leshchenko⁴, Kseniia Dukhnovska⁵, Anatoliy Biehun⁶

¹Doctor of Technical Sciences, Professor, Head of Department of Mathematics, State University of Telecommunications, Ukraine, bar64@ukr.net

²Doctor of Technical Sciences, Professor, Department of Information and Cybersecurity Systems, State University of Telecommunications, Ukraine, alaptev64@ukr.net

³ PhD in Physics and Mathematics, Associate Professor, Department of Software Systems and Technologies, Taras Shevchenko National University of Kyiv, Ukraine, kovok@ukr.net

⁴ PhD in Technical Sciences, Associate Professor, Department of Networking and Internet Technologies, Taras Shevchenko National University of Kyiv, Ukraine, lesolga@ukr.net

⁵Assistant of the Department of Applied Information Systems, Taras Shevchenko National University of Kyiv, Ukraine, duchnov@ukr.net

⁶ PhD, Professor, Director of the Institute of Information and Communication Technologies and Systems, Vadym Hetman Kyiv National University of Economics, Kyiv, Ukraine, begunt@ukr.net.

ABSTRACT

The article considers mathematical models of text documents. Currently, there is an exponential increase in the amount of information both in electronic repositories and in real life. An important element of the study is the dynamics of information. The dynamic characteristics of information should be taken into account in search engine algorithms. In this work, we built a dynamic model of a text document based on TF-IDF measures and conducted experiments to apply it in search algorithms.

The software implementation of the experiments was carried out in Python 3.8 in the Spider environment. The simulation results showed that the proposed dynamic model of the TF-IDF method can improve the accuracy of the search for text documents.

Key words: Dynamic model, tf-idf measure, text mining, dynamics of information, parameters, information.

1. INTRODUCTION

The concept of information unites many branches of science. Information is the object of research in informatics, systems analysis, mathematics, in physics, and so on. Today, there is an exponential growth in the amount of information both in electronic storages and in real life. According to the statistics of the website www.worldwidewebsite.com, as of February 2020, about 61 billion pages are registered on the Internet, in which some information is displayed. Every year and a half, according to the research of the international analytical company IDC, the amount of information on the Internet doubles. At the same time, its quality is changing: the content on existing resources is being specified, in order

to attract the attention of a larger audience. From here an important element of research is the dynamics of information.

1.1 Literature analysis and problem statement

In the course of researching information, the following properties are distinguished: relevance, timeliness, completeness and accuracy. The relevance of information is the relevance of its importance at the time when it entered. The timeliness of information characterizes the usefulness of information regarding its application to the solution of applied problems at the moment when this problem is to be solved. If there is enough information to solve a specific problem, then they say that the information is complete. The correspondence of information about an object, process or phenomenon to their present state determines its accuracy [1,2,22].

Information is exchanged in various ways; it can be represented in many formats. This article deals only with its textual representation [3,4,23].

A text document in this article means the information presented on any material carrier in textual form.

A text document is a static object. But the collection of documents is a dynamic object, as it reflects the evolution of some topics. Collections of volumes of scientific conferences, thematic sites, posts in social networks are examples of dynamic collections of text documents. In them, text documents or messages are defined in epochs.

The increase in the physical volumes of textual information and the dynamics of their content flows leads to the emergence of information noise, duplication of textual information, and overflow of information. The generalization of large dynamic streams of content that are continuously generated in Internet sources requires qualitatively new approaches for constructing models of text documents [5 -

7,24]. In this case, the input information is the text in natural language as a sequence of terms. For the content analysis of dynamic collections, it is necessary to use a dynamic model of a text document.

1.2 Aim of the article

The purpose of this article is to develop a dynamic model for selecting the necessary information from a collection of documents.

Conduct computer simulations in the MATLAB environment according to the proposed method in order to evaluate the effectiveness of the proposed method.

2. THE MAIN SECTION

The following standard hypotheses are used to build models of text documents and their collections:

- the order of the documents in the collection does not matter;
- the order of terms in the document does not matter;
- words that appear in most text documents do not matter;
- Terms in different forms are one and the same word.
- In order to make possible the computer analysis of text documents, the texts are pre-processed. The main stages of which are as follows:
- reduction of all words to normal form (lemmatization or stemming);
- highlighting terms and highlighting key phrases;
- deleting stop words.

Under the text understand the finite set of words, united lexical, grammatical, meaningful and frequency ratio, which forms an informational message.

Figure 1 shows the process of accumulating text documents in a collection of documents.

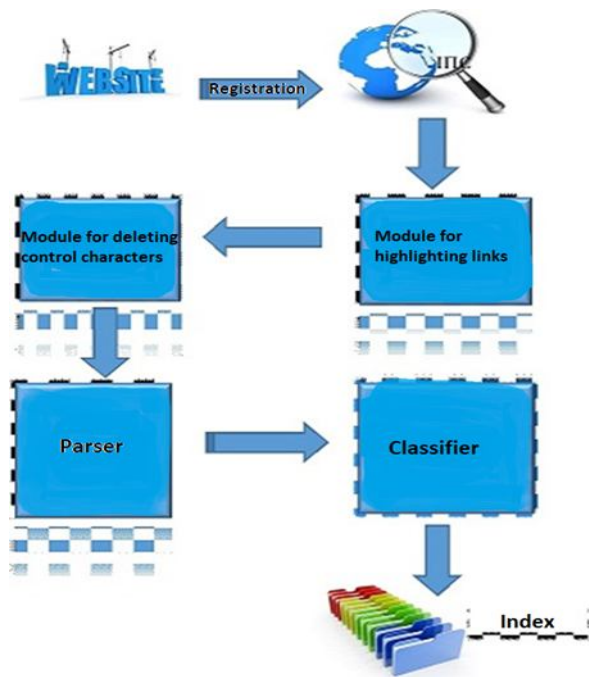


Figure 1: Preliminary processing of text documents

A collection of text documents will be understood as any set of text documents located on a local computer, server, cloud or search engine index.

First of all, the manufacturer of text documents must register their resource in the electronic storage. After registration, the document enters the module to highlight the links. Links from the current document are sent to the queue for downloading a new document via this link. Next, the document enters the module, in which control characters, commands, etc. are deleted, that is, all invisible components of the document. The output is a clear text without all the excess, which is passed to the parser (special

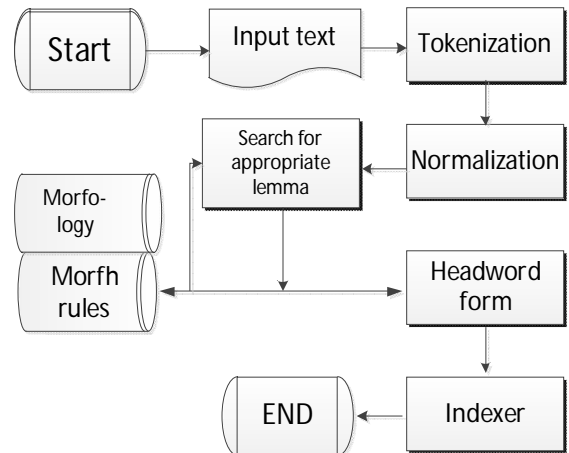


Figure 2: Lemmatization

module). The functions of the parser include the parsing of the text and the selection of terms from the text of the document. This module calculates all metrics necessary for analyzing a document and building its search image. Then the document, or a set of separate terms from it, is submitted to the indexer, which classifies the document, that is, finds a suitable place in the categories of electronic storage - the index, and records in the appropriate format.

Each text document is a collection of terms that carries some information. A term or a word is a syntactically independent complex of morphemes forming a rigidly connected structure. A word differs from a word combination in that some of its elements cannot be used independently. Component words have tougher connections than component sentences. Lemmatization and stemming algorithms are used to account for all word forms of a single word.

Lemmatization (Fig.2) is the process of reducing the different forms of a word to one single form in accordance with the grammatical forms of a particular language [8].

The stemming is a process of linguistic normalisation, in which use to the removal of derived affixes. To date, there are many different algorithms for stemming. Particularly notable among them are: Porter's Stemmer, KSTEM algorithms, and n-grams. Porter's Stemmer is an algorithm that does not use word vocabulary. It applies a number of rules by which affixes are cut off, basing from the grammar

of the language. Porter’s Stemmer works fast, but not unmistakably [9-10].

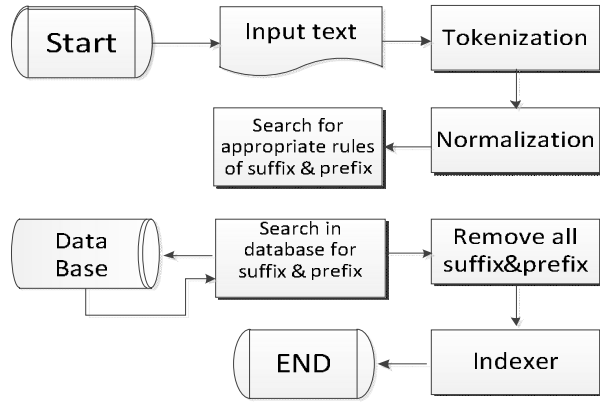


Figure 3:Stemming

The *KSTEM* algorithm is a morphological analyzer whose work is based on the algorithm for replacing the suffix and searching for the base word from which the source word originated (Fig.3). The n-gram algorithm is based on the postulate: “If word A coincides with word B, taking into account several errors, then with a high degree of probability they will have at least one common substring of length N”. Such substrings whose length is n are called n-grams. At the time of parsing, the word is broken down into n-grams, and then the word falls into lists for each of these n-grams. When searching, the query is also divided into n-grams, and each of them is sequentially searched through the list of words containing the given substring [11-14,21].

2.1 Text document models

In the simplest case, only the fact of the presence or absence of a term in a document can be considered in a textual analysis. This document model is called binary. A complication of this model is the approach, where for each term not only its presence is indicated, but also some of its “weight”. In this case, the weight can be assigned to words, phrases, or to the basics of words [15-20].

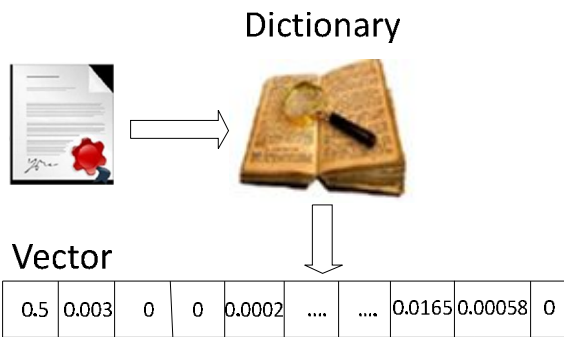


Figure 4:Vector text model

The simplest method of weighing words in a document is the method of counting the number of occurrences of a term (term weight) in a document (Fig. 4). In this case, it is considered that if a term occurs more frequently in the text

of a document, then this document is more likely to be related in content to this term. The disadvantage of this method is that documents of greater length may have a greater weight of the terms included in them. At present, the most common method is to calculate the frequency of the occurrence of terms in a document (TF). Frequency is the ratio of the number of occurrences of a term in the text of a document to the total number of terms of this text. The disadvantage is that here, on the contrary, long documents are underestimated, since they contain more terms, and the average frequency of the terms in the text below. To combat this effect is applied normalized frequency, which is calculated as $0.5 + 0.5(TF/NTF)$, where *NTF* is the average frequency of the term in the document. An alternative method of weighting terms in the texts of the document is the logarithm of the frequency of occurrence of the term. In this case, the weight of terms included in the text of the document is defined as $1 + \log(TF)$. To compensate for the effect of different resource lengths, a similar frequency normalization is used. In this case, the formula looks like

$$\frac{1 + \log(TF)}{1 + \log(MTF)}$$

that occurs the most times in the document.

These methods for determining weight are well described in [5-7].

Today, a text document model based on the static measure TF-IDF is widely used.

Suppose we have a dictionary W - an ordered set of terms whose power is M. The power of a dictionary is the number of terms it contains. Then the document can be represented as a vector:

$$D_i = \langle w_{i1}, w_{i2}, \dots, w_{iM} \rangle, \quad (1)$$

where w_{ki} is the frequency of the k-th term in the i-th text document ($i = (1, n)$).

$$TF_{ki} = \frac{m_{ik}}{M_i}, \quad (2)$$

m_{ki} is the number of occurrences of the k-th term in the i-th document;

M_i — the total number of terms in the i-th document;

$$IDF_k = \ln \frac{N}{n_k} \quad (3)$$

N is the total number of documents in the collection, n_k is the number of documents in the collection in which the k-th term is found. The value IDF_k characterizes the importance of the k-th term in the collection of documents. The frequency of the term is calculated using the TF-IDF formula:

$$w_{ki} = TF_{ki} * IDF_k \quad (4)$$

2.2 Dynamic characteristics of text documents

Over time, the information grow old, i.e. loses its value and utility. The reason for this is the constant appearance of new

documents that contain more relevant, more complete, more accurate and reliable information.

When constructing algorithms for text analysis, it is necessary to take into account the dynamic characteristics of the information, since this contributes to the determination of the greater significance of relevant documents.

The process of aging information for different subject areas or even topics occurs at different speeds. The rate of change of mathematical patterns, for example, differs from the rate of development of information technology algorithms. Many things influence this process. Features of this process are related to the development prospects of each of the topics. In 1960, R. Barton and R. Kebler made a study and determined the periods of life of publications on physics - 9.2 years, on mathematics - 21, geology - 23.6 [21-22]. By analogy with quantum mechanics, researchers introduced the concept of "half-life" of scientific articles. The "half-life" of scientific articles is the period during which half of all scientific articles of the chosen topic are published.

To obtain a quantitative estimate of the rate of aging of scientific publications R. Barton and R. Kebler used the Malthus model. This model may occur under the following assumptions:

- there is N - the number of scientific papers on some topics;
- it is assumed that the growth rate of the number of scientific works is directly proportional to N.

The latter assumption follows from the statistical studies of the international research firm IDC.

If we consider the model of the document, which is given by formulas (1-4), then one of the components of this model TF does not depend on time, provided that the number of words in the dictionary does not change with time. Indeed, TF is calculated according to formula (2), but neither the number of occurrences of the k-th term in the i-th document, nor the total number of terms in the i-th document change over time. The dynamic component of this model is the IDF value, which is calculated according to formula (3).

Let $IDF_k(t)$ be the importance of the k-th term in the collection of documents at time t on this topic. The relative change of this quantity over time Δt is defined as

$$R(t) = \frac{IDF_k(t + \Delta t) - IDF_k(t)}{IDF_k(t) * \Delta t} \quad (5)$$

Then

$$R(t) * IDF_k(t) = \frac{IDF_k(t + \Delta t) - IDF_k(t)}{\Delta t} \quad (6)$$

in passing to the limits we get:

$$R(t) * IDF_k(t) = \frac{dIDF_k(t)}{dt} \quad (7)$$

As a result, we obtained a differential equation with separable variables. Its solution depends on the function R(t), i.e. from the relative change in the importance of the k-th term in the collection of documents. Integrating the left and right side of the equation:

$$\int_0^t \frac{dIDF_k}{IDF_k} = \int_0^t R(t) dt \quad (8)$$

$$\ln(IDF_k) = \int_0^t R(t) dt \quad (9)$$

$$\text{or } IDF_k(t) = e^{\int_0^t R(t) dt}$$

If R (t) is independent of time, i.e. is a constant, then we get the Malthus model:

$$IDF_k = IDF_{k0} e^{\alpha_c t - \beta_c t} \quad (10)$$

where:

- IDF_{k0} - the importance of the k-th term at time t = 0;
- β_c - the half-life of the relevance of the document related to the subject C, is determined by an expert way, for each subject separately;
- α_c - growth rate of the relevance of the subject documents C;
- C- Subject text document.

In support of the above, modeling was carried out to determine

dependence of the K-th element of information on the rate of its growth and the "half-life"

The simulation results are shown in Figure 5.

The results obtained fully confirm theoretical calculations. with the growth of relevance, depending on the decrease in search options, the dependence of the detection of the K-th element of search information increases.

Dependence of the K-th element of information search on its actualization

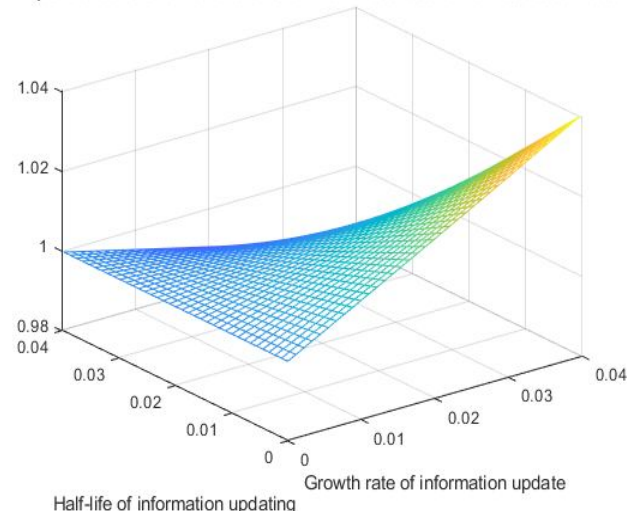


Figure 5: Dependence of the K-th element of information on its growth rate and the "half-life" of information.

The advantage of this model is that the Malthus equation has an exact solution in the form of a simple and convenient exponential function, but from the point of view of interpreting the results, it looks rather dubious. The main disadvantage of this approach is that the exponential function cannot describe events that have local extremes, but for a

large number of text documents, the Malthus model is correct.

Consider a model (1), where the weight of the term w_{ik} is defined by formula (4). This formula is the product of the stationary component of the TF and the dynamic IDF. Then relying on the Malthus model, you can get:

$$w_{ik}(t) = TF_{ik} IDF_{k0} e^{\alpha_c t - \beta_c t} \quad (11)$$

Thus, a dynamic model of a text document is obtained.

2.3 Results

To compare the models of text documents (4) and (5), the problem of searching for text information was considered.

2.3.1 Statement of the problem of searching for textual information

Let a collection of textual documents be given, you need to find documents in it upon request on a given topic. The collection of documents is presented in the form of a matrix (1), the rows of which are vectors constructed using the model (4), or (5). A request is a short text document and can be represented as a vector (4) or (5).

Often, the same methods are used in the search engine search module as in the indexing module. For example, in the vector model, the search is built on the basis of the tuple $\langle W, L_D, S, Q \rangle$, where W is the dictionary, that is, an ordered set of terms, the power of the dictionary is denoted as N ; information flow is a matrix: $L_K = D_i$ of dimension $K \times N$, where search images of K documents. Similarly, they present a vector to the query: $Q = \{q = \langle t_1, t_2, \dots, t_N \rangle\}$. Hence, the search procedure S has the form $Lxq=r$, where q is the query vector, r is the system response to the query.

2.3.2 Decision algorithm

In order to search for information on a given query, you need to sort through all the documents in the collection and calculate the distance between the vector representing the collection document and the vector representing the query. The smaller the distance between the document vector and the request vector, the more this document corresponds to the request:

$$r_k(t) = \sqrt{\sum_{i=1}^n (w_{ik} - w_{iq})^2} \quad (12)$$

where r_k - is the distance from the vector that represents the k -th document to the vector that represents the request,

w_{ik} - the coordinates of the vector that represents the k -th document,

w_{iq} - the coordinates of the vector that represents the request.

Further, the documents are sorted, depending on the distance to the request and the most relevant documents are selected.

Search quality characteristics are divided into two error levels. A first level error is considered if the document is not

mistakenly among the documents sought. Errors of the second level include errors when a document is mistakenly found in the searched documents. Let the number of documents in the test set be equal to N , of which N_p is the number of documents matching the request, and N_n is the number of documents that are not related to the request.

Then, $N = N_p + N_n$.

Let the number of false passes F_n , and false detections F_p , therefore the number of correct passes and correct detections: $T_p = N_p - F_n$; $T_n = N_n - F_p$. The degree of accuracy and completeness that are often used in information retrieval tasks is calculated based on the characteristics of T_p and F_p :

$$\begin{aligned} precision = P &= \frac{T_p}{T_p + F_p} \times 100\%, \\ recall = R &= \frac{T_p}{T_p + F_n} \times 100\% \end{aligned} \quad (13)$$

Completeness measures the proportion of correct information across all documents. Accuracy measures the proportion of true detections of all identified resources. Completeness and accuracy are values dependent on each other. During the development of search engine architecture, one usually has to choose one of two characteristics as dominant. If the choice fell on accuracy, this leads to a decrease in completeness due to an increase in the number of false-positive answers. The increase in completeness causes a simultaneous drop in accuracy. Therefore, it is convenient to characterize the search engine using one value, the so-called F_1 -measure or Van Riesbergen measure.

$$F_1 = 2 \frac{P \times R}{P + R} \quad (14)$$

Measure F_1 is one of the most common characteristics for such systems. There are two main approaches to calculating F_1 for text document search tasks: total F_1 (the results for all tests are summarized in one table, by which the measure F_1 is then calculated) and the average F_1 (for each test its own F_1 value is generated, then the arithmetic average for all tests).

The percentage of errors allows you to determine the correctness metric:

$$A = \frac{T_p + F_p}{N} \quad (15)$$

As a working material for the experiments, a test sample of 400 text documents in Ukrainian was taken, among which there were 250 texts on continuum mechanics (the aging coefficient of information in physics is 4.6). Requests were fulfilled for documents on continuum mechanics.

The software implementation of these experiments was carried out in Python 3.8 in the Spider environment. Work with files: the authors using regular expressions and additional libraries (re - regular expressions, pynlpl - word processing, pandas - for data manipulation and analysis

implement opening text documents and removing unnecessary meta-information). For lemmatizing text terms, we used the pymorphy2 library, which was created to process the Russian language, but also contains Ukrainian dictionaries for morphological analysis. The pymorphy2 library also provides the ability to remove stop words from document text. Stop-words are words that do not carry the semantic load of the text and are found in every text document.

Because of the work done, the following results were obtained:

Table 1: Result of the work

Text document model	Accuracy	Completeness	F1 measure	Error percent
TF-IDF	72,9%	74,83%	73,85	0,49
Dynamic TF-IDF	73,65%	72,06%	73,9	0,48

3.CONCLUSION

The dynamic characteristics of the information should be taken into account when searching for text documents, because if you need documents in which the necessary information should work and will be used for making decisions or for other control actions, in this case the time for creating the information should be taken into account. Otherwise, the decision or guidance may be ineffective.

In the work, the constructed dynamic model of a text document based on TF-IDF can be used in information retrieval systems. In this case, the sorting of documents and the proximity between them can be considered on the basis of the Euclidean distance. Unfortunately, if we consider the sinusoidal distance, then using the dynamic model of the information resource, it will not change, because as a result of applying this approach, we get a collinear source vector. The way out is to replace the vector model of information resources with the eTVSM model - a thematic vector model in which a document or information resource is considered as an association of smaller documents or information resources, but on different topics [23-24]. Considering that different subjects vary in time with different speeds, we will not get the vector collinear to the original.

Using various algorithms for classifying text documents based on the developed model, the tasks of automating the construction of a classifier can be solved. At the same time, time accounting in the model can clarify the place of a text document in the classifier.

The search for text documents based on the Malthus model has an exact solution in the form of a simple and convenient exponential function, but from the point of view of interpreting the results, it has several disadvantages. The main disadvantage is that the exponential function cannot describe events with local extremes, but for a large number of text documents, the Malthus model is correct.

A dynamic model of a text document based on the TF-IDF method is constructed. In addition, modeling was carried out

with a series of experiments that allowed drawing conclusions about the effectiveness of the proposed method and its application in information retrieval systems.

REFERENCES

1. Perera,R. **Recent Advances in Natural Language Generation: ASurvey and Classification of the Empirical Literature.** *Computing and Informatics*, 2017, vol. 36, pp. 1–32.
2. Kravchenko Y. **Evaluating the effectiveness of cloud services.** *2019 IEEE 1th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S&T'2019, Kyiv*, 2019, pp.120–124
3. Pennington, J.; Socher, R., **Manning, C. D. Glove: Global vectors for word representation.** *EMNLP*, 2014, vol. 14, pp. 1532–43.
4. Kiros, R., Zhu Y., Salakhutdinov R. R. **Skip-thought vectors.** *Advances in Neural Information Processing Systems*, 2016, pp.3276– 3284.
5. Dukhnovska K.K. **Formuvannya Posukovy dynamical vector space.** *Shtunniy intertekt*, 2016, no3.4, pp. 28–36.
6. Barabash O.V., Open'ko P.V., Kopyika O.V., Shevchenko H.V., Dakhno N.B. **Target Programming with Multicriterial Restrictions Application to the Defense Budget Optimization.** *Advances in Military Technology*, 2019, vol. 14, no. 2, pp. 213 – 229 (2019).
7. Kreines M.G., Kreines E.M. **Control model for the alignment of the quality assessment of scientific documents based on the analysis of content-related context.** *JCSSI*, 2016, vol. 55, no. 6, pp. 938–947
8. Musienko A.P., Serdyuk A.S. **Lebesgue-type inequalities for the de la Vallée-Poussin sums on sets of analytic functions .** *Ukrainian Mathematical Journal* September 2013, Volume 65, Issue 4, pp. 575 – 592.
9. Musienko A.P., Serdyuk A.S. **Lebesgue-type inequalities for the de la Vallée poussin sums on sets of entire functions .** *Ukrainian Mathematical Journal* October 2013, Volume 65, Issue 5, pp. 709 – 722.
10. Grigoryan D.S. **Cogherent data processing in tasks of spectral analysis of super resolution radar signals.** *Journal of Radio Electronics" Electronic Journal*. 2012. № 3 <http://jre.cplire.ru/jre/mar12/1/text.html>.
11. Ara Jullion A. Abello, Gabriele Francesca Y., Domingo, Maria Jamelina T. Joven, Samanta Alexis S. Malubay. **Power Measurement Model Optimization using MATLAB.** *International Journal of Advanced Trends in Computer Science and Engineering. (IJATCSE)*. 2019. Vol. 8, № 3, May – June. pp. 538 – 542.
12. Bakiko V.M., Popovich P.V., Shvaichenko V.B. **Determination of noise immunity of a communication channel in case of accidental interference.** *Bulletin of the National tech. University "KhPI"*: Coll. Science. Kharkiv: NTU "KhPI", 2018. № 14 (1290). P. 7 - 10.
13. Milov O., Yevseiev S. Milevskiy S. Ivanchenko Y., Nesterov O., Puchkov O., Yarovyi A., Saliy A., Tiurin V., Timochko O. **Development the model of the**

- antagonistic agent's behavior under a cyber-conflict.***Eastern European Journal of Advanced Technologies.* Kharkiv.2019. 4/9 (100). pp. 6–19.
14. Lubov Berkman, Oleg Barabash, Olga Tkachenko , Andri Musienko, Oleksandr Laptiev, Ivanna Salanda.**The Intelligent Control System for infocommunication networks.***International Journal of Emerging Trends in Engineering Research (IJETER)* Volume 8. No. 5, May 2020. pp.1920 – 1925.
15. Savchenko Vitalii, Syrotenko Anatolii, Shchypanskyi Pavlo, Matsko Oleksander, Laptiev Oleksander.**The Model of Localization Precision for Detection of Hidden Transmitters.** *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Volume-9 Issue-4, February 2020. ISSN: 2278-3075. pp. 2114-2119.
16. Olexandr Laptiev, German Shuklin, Spartak Hohonianc, Amina Zidan, Ivanna Salanda.**Dynamic model of Ceber Defence Diagnostics of information Systems with the Use of Fozzy Technologies***IEEE ATIT 2019 Conference Proceedings Kyiv, Ukraine, December 18-20, pp.116-120.*
17. Sweta Srivastav, Sangeeta Gupta.**Results with Matlab coding of Middle Graph of Cycle and its related graphs in context of Sum Divisor Cordial***International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Volume-8 Issue-2, February 2020. pp.398-401.
18. Aaron Don M. Africa, Lourdes Racielle Bulda, Matthew Zandrick Marasigan, Isabel Navarro. **Binary Phase Shift Keying Simulation with MATLAB and SIMULINK***International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Volume-8 Issue-2, February 2020.
19. Mashkov O.A., Sobchuk V.V., Barabash O.V., Dakhno N.B., Shevchenko H.V., Maisak T.V. **Improvement of variational-gradient method in dynamical systems of automated control for integro-differential models.** *Mathematical Modeling and Computing*, 2019, Vol. 6, No. 2, pp. 344 – 357.
20. Barabash O., Dakhno N., Shevchenko H., Sobchuk V. **Integro-Differential Models of Decision Support Systems for Controlling Unmanned Aerial Vehicles on the Basis of Modified Gradient Method.***IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC).* 16-18 October, National Aviation University, 2018. Kyiv, Ukraine. pp. 94 – 97.
21. Ihor Ruban, Nataliia Bolohova, Vitalii Martovytskyi, Nataliia Lukova-Chuiko , Valentyn Lebediev. **Method of sustainable detection of augmented reality markers by changing deconvolution.***International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE).* Volume 9, No.2, March-April 2020, pp.1113-1120.
22. Laptiev Oleksandr, Shuklin German, Savchenko Vitalii, Barabash Oleg, Musienko Andrii and Haidur Halyna. **The Method of Hidden Transmitters Detection based on the Differential Transformation Model.** *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE).* Volume 8, No 6, November – December 2019, pp. 2840 – 2846.
23. W. N. Hussein, L. M. Kamarudin, M. R. Hamzah. , H. N.Hussain, K. J. Jadaa. **A Methodology for Big Data Analytics and IoT-Oriented Transportation System for future implementation.***International Journal of Emerging Trends in Engineering Research (IJETER).* Volume 8, No 11, November .2019, pp. 449 – 453.
24. Varun K. L. Srivastava, N. Chandra Sekhar Reddy, Dr. Anubha Shrivastava.**An efficient Software Source Code Metrics for Implementing for Softwarequality analysis** *International Journal of Emerging Trends in Engineering Research (IJETER).* Volume 7, No 9, September .2019, pp. 216 – 222.