

ISSN 2663-3051
(ISSN 0555-2656 до 2019 р.)

БІОНІКА ІНТЕЛЕКТУ

ІНФОРМАЦІЯ, МОВА, ІНТЕЛЕКТ

№ 1 (98)

2022

НАУКОВО-ТЕХНІЧНИЙ ЖУРНАЛ

Заснований у жовтні 1967 р.

Засновник та видавець
Харківський національний університет радіоелектроніки

Періодичність видання – 2 рази на рік



Науково-технічний журнал
«БІОНІКА ІНТЕЛЕКТУ»

ISSN 2663-3051

Заснований Харківським національним університетом
радіоелектроніки у 1967 році

Реферування та індексування:

Google Scholar



INDEX  COPERNICUS
I N T E R N A T I O N A L



Журнал включено до списку наукових спеціалізованих видань України
з технічних та фізико-математичних наук
згідно з наказом Міністерства освіти і науки України № 820 від 11.07.2016



V. A. Shyrovok

Ukrainian Language Information Foundation
of the National Academy of Sciences of Ukraine*Dedicated to Sir Roger Penrose –
the outstanding intellectual
of our time.***THE “CORONA” CRISIS AND THE “DARK MATTER” OF DOLLAR**

The ideas and methods of the general theory of evolution formulated by the author in previous publications have been developed in the work. The provisions of the author's concept are systematized, which are presented in the form of three laws of evolution concerning the essence and features of the functioning of the main concept of the theory – *the mechanism of evolution*.

The First (basic) law of evolution states that the complexity of an evolving system always increases. Thus, an increase in the complexity of one or another mechanism of evolution is a necessary condition for evolution, in the area where this mechanism operates. The Second Law of Evolution defines the qualitative nature of the behavior in time of any mechanism of evolution, that is, its temporal dynamics. When formulating the Third Law of Evolution, it is noted that the increase in complexity is only a necessary, but not a sufficient condition for evolution. Under some circumstances, the system gets the opportunity to demonstrate an abnormal increase in complexity, while practically not evolving. As a result, a situation arises of an uncontrolled, "parasitic" increase in the complexity of the system, not directly related to its evolution, which gives reason to single out a parasitic component in the complexity of the system, which gives a constantly increasing load on the system. This is the basis for formulating the Third law of evolution – the law of growth " parasitic" complexity, which explains the essence and causes of almost all evolutionary catastrophes.

The laws of evolution are illustrated by examples of the behavior of the genetic, neural, communication and financial-information-network mechanisms of evolution. Of particular interest is the evolutionary interpretation of the inflationary model of the Universe in the approach of Roger Penrose. In the spirit of the above conceptual provisions, an analysis was made of the crisis associated with the COVID-19 pandemic ("Corona" Crisis"), which was used by the ruling circles as an element of "controlled chaos". Having produced a global disturbance of the world socio-economic system, the crisis remained manageable to an acceptable extent. At the same time, the World Financial System with the dollar at the head has withstood and, perhaps, even somewhat strengthened.

An analysis of all the mentioned phenomena and processes leads to the conclusion that in any system at a certain stage of evolution, a structure begins to appear, consisting of three zones: controlled (managed); uncontrolled (unmanaged) and intermediate between them – weakly (conditionally) controlled (managed). In terms of observability, it sounds like any sufficiently complex system has an observable ("light") part, an unobservable (dark"), and an intermediate ("gray") region between them. The noted effect is universal and manifests itself in almost all subject areas.

This approach allows, within the framework of a unified methodology, to analyze the phenomena and processes of both natural and social history. The "Corona" Crisis, as a representative of a series of acute global crises of our time, really turns out to be connected with the "Dark Matter" of the financial-information-network mechanism.

The author draws attention to the fact that crises and catastrophes occupy a very significant place in the arsenal of Evolution's tools. This circumstance poses very responsible tasks for the scientific community.

Широков В.А. Криза «Корони» і «Темна Матерія» долара. У роботі набули розвитку ідеї та методи загальної теорії еволюції, сформульованої автором у попередніх публікаціях. Систематизовано положення авторської концепції, які представлені у формі трьох законів еволюції, що стосуються сутності та особливостей функціонування основного концепту теорії – *механізму еволюції*.

Перший (основний) закон еволюції стверджує, що складність системи, що еволюціонує, завжди зростає. Таким чином, зростання складності того чи іншого механізму еволюції є необхідною умовою еволюції на тій ділянці, де діє даний механізм. У Другому законі еволюції визначається якісний характер поведінки у часі будь-якого механізму еволюції, тобто його темпоральна динаміка. При формулюванні Третього закону еволюції зазначається, що зростання складності є лише необхідною, але не достатньою умовою еволюції. За деяких обставин система має можливість демонструвати аномальне зростання складності, при цьому практично не еволюціонує. В результаті виникає ситуація неконтрольованого, «паразитного» збільшення складності системи, не пов'язаного безпосередньо з її еволюцією, що дає підставу виділити у величині складності системи паразитну компоненту. Це дає навантаження на систему, яке постійно зростає. Викладене дає підстави для формулювання Третього закону еволюції – закону зростання «паразитної» складності, який пояснює суть та причини практично всіх еволюційних катастроф. Закони еволюції проілюстровані на прикладах поведінки генетичного, нейронного, комунікаційного та фінансово-інформаційно-мережевого механізмів еволюції. Особливий інтерес викликає еволюційна інтерпретація інфляційної моделі Всесвіту у підході Роджера Пенроуза. У дусі викладених концептуальних положень виконано аналіз кризи, пов'язаної з пандемією COVID-19 («Кризою «Корони»»), використаної правлячими колами як елемент «керованого хаосу». Спричинивши глобальне збурення світової соціально-економічної системи, криза залишалася прийнятною мірою керованою. При цьому встояла і, можливо, навіть дещо зміцніла Світова фінансова система з долларом на чолі.

Аналіз усіх згаданих явищ і процесів веде до висновків, що в будь-якій системі на певному етапі еволюції починає проявлятися структура, що складається з трьох зон: контрольованої (керованої); неконтрольованої (некерованої) та проміжної між ними – слабко (умовно) контрольованої (керованої). У термінах спостережуваності це звучить так, що будь-яка досить складна система має спостережувану («світлу») частину, крім того, частину яка не спостерігається (темну), і проміжну між ними («сіру») область. Зазначений ефект є універсальним і проявляється практично у всіх предметних галузях. Цей підхід дозволяє в рамках єдиної методології виконувати аналіз явищ і процесів як природної, так і соціальної історії. «Криза «Корони», як представник низки гострих глобальних криз сучасності, справді виявляється пов'язаною з «Темною Матерією» фінансово-інформаційно-мережевого механізму.

Автор звертає увагу, що в арсеналі інструментів Еволюції кризи та катастрофи займають дуже важливе місце. Ця обставина ставить перед науковою спільнотою дуже відповідальні завдання.

INTRODUCTION

On March 19, 2021, at the Kyiv House of Scientists of the National Academy of Sciences of Ukraine, located on Volodymyrska Street, number 45-a, in a beautiful mansion of the early 20th century, the author of this article, at the request of the Kyiv scientific youth, gave a lecture under that title. As everyone remembers, this happened in the era of COVID-19, when, after its apotheosis in 2020 (the peak of the «corona crisis»), the authorities were just beginning to ease pandemic restrictions. This was done very carefully and rather inconsistently, which was dictated by the great uncertainty of this global phenomenon that engulfed the whole world, and a poor understanding of its nature, causes and methods of dealing with it.

Be that as it may, the author, following all the prescribed restrictions and precautions, gave this lecture, was gladly subjected to the temperamental attacks of young scientists and strengthened in the thought of the need to publish a fundamental work, most likely a book on the topics of evolution, a topic that served as a logical and the factual basis of the lecture. Before that, I had repeatedly raised the evolutionary topic and made reports on it for completely different audiences, including at the meetings of the N. M. Amosov Club, of which the author has the honor to be a member, at the House of Scientists.

Time passed, but the process of writing the book slowed down. Much more pressing matters arose, dealing literally with matters of life and death. Analyzing these dramatic events, I became more and more deeply convinced that they are quite consistent with the evolutionary doctrine I am developing, the first publications of which were carried out several years earlier in the journal “Bionics of Intelligence”¹. And finally, at the suggestion of professor G. Chetverikov, through whose efforts the journal only continued to exist in recent years, I found time and wrote the text of the article with the announced title, which I offer to the attention of readers.

Before starting the presentation of the main content, I consider it necessary to make some clarification. The fact is that for many, it may seem artificial and not intelligible

1 В.А. Широков. Эволюция как универсальный естественный закон (Пролегомены к будущей общей теории эволюции). «БИОНИКА ИНТЕЛЛЕКТА», №№ 1-2, 2017, №1, 2018. В свободном доступе по адресу: <https://ulif.org.ua/page/evoliutsiia-kak-universalnyi-iestiestviennyi-zakon-prolieghomienny-k-budushchiei-obshchiei-tie>

enough to compare such different things as the global pandemic and the matter of the dollar, moreover, dark (this epithet can also cause perplexing questions). However, below the author will try to present the phenomenology and logic of this case in such a way that this comparison would look not only quite rational, but even necessary.

To facilitate understanding of the content, let me quote here the main provisions of the work cited above on the general theory of evolution. Let's start with this.

Part I. EVOLUTION AS A UNIVERSAL NATURAL LAW

1.1. General Provisions and Fundamental Issues

We proceed from the conviction that the World is evolving. We qualify this belief as the “Presumption of Development” and further we will consider evolution as one of the main properties of the World, perhaps even the main one in general. There have been many attempts in history to give a “strict” definition of the concept of “*evolution*”. We believe that “*evolution*” is such a fundamental, primary concept that it is impossible to give a strict definition of it. However, you can give examples, give some explanations, explore its properties and establish patterns. But, when it comes to regularities, it is subject to clarification what exactly is meant. In our opinion, this clarification is concluded in the answer to two questions that we consider fundamental for the subject of our consideration:

1. *If we analyze evolving objects, then: is it possible to set those properties that can be correlated with any process of evolution?*
2. *If it is possible to set these properties, then: are they subject to generalization and/or are they reducible to some “general” concept that could be associated with any evolutionary process?*

The author claims that there is a positive answer to both questions. To answer them, we define the conceptual basis of our theory.

1.2. On the mechanisms of evolution.

We believe that the basic concept of the general theory of evolution is the concept of the “*Mechanism of evolution*”. The brief content of this concept is revealed in the following statements:

1. The evolution of the world is carried out in certain discrete forms. These discrete forms we will call “mechanisms of evolution”.

2. Each mechanism of evolution is characterized by a certain complex of materially expressed features, specific to it, that allow it to be individualized and distinguished from other mechanisms as some discrete entity².

3. However, all mechanisms have some common properties, which allows them to be typified and attributed to one class of phenomena, namely, to the class "Evolution"

In the cited work, the following evolutionary mechanisms are singled out and considered with varying degrees of detail:

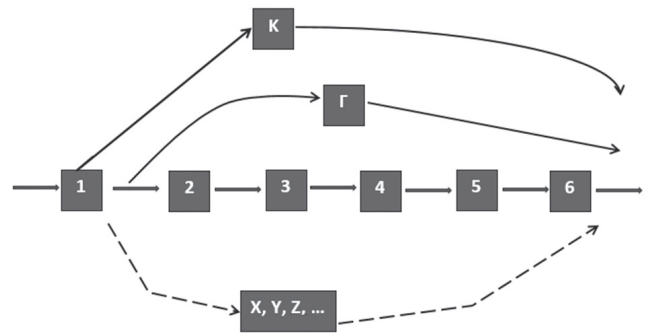
1	PHYSICAL	E	C	O	G	N	I	T	I	V	E
2	CHEMICAL										
3	GENETIC										
4	NEURAL										
5	COMMUNICATION										
6	PRODUCTION, consisting of the stages:										
6.1.	Consumer	Economic	?	?	?	?	?	?	?	?	
6.2.	Exchange										
6.3.	Financial										
6.4.	Financial Information										
6.5.	Financial-Information-Network										
?	??	?	?								

Note that each of these mechanisms has its own "fine structure". Here we have explicitly listed only individual stages of the Production Mechanism, of which we currently number five and some which we plan to dwell on in more detail. At the first glance at the table, one notices that the last line consists of only question marks. This means that at the moment it remains unclear: how will the Financial-Information-Network Mechanism end?, what will follow it?, how and when will the transition to the next mechanism be carried out?, what will it be like? and what will happen to the human race at that time (if it exists at all by that time)? Unfortunately, we do not know clear answers to these questions, although there are a great many futurological fantasies on these topics.

Does the above list exhaust the set of all mechanisms of evolution? The answer is obviously no. We assume that this set is generally open. Moreover, the possibility of parallel action and interaction of several mechanisms, and not only those listed in the table, is by no means excluded.

2 The principle of such individualization and selection, apparently, is close to discovered and described by us the lexicographic effect in information systems (see. В.А.Широков. Інформаційна теорія лексикографічних систем. – К. : Довіра, 1998. – 331 с. V. Shyrokov. "Language. Information. System" –К.; Наукова думка., 2021, 158 с.. Free access at: https://www.ulif.org.ua/system/files/blok-shirokov_book.pdf).

So, the following diagram shows a picture where, in parallel with the six mechanisms we have designated (squares 1 – 6), the cosmic (K), geological (G) and a number of other possible others (X, Y, Z, ...) operate.



Each mechanism of evolution operates on a certain class of objects, and its action is provided by the corresponding "substance", a kind of "agent" system that implements the action of this mechanism. The noted facts are summarized in the table.

MECHANISM OF EVOLUTION	OBJECTS	«SUBSTANCES»
Physical	Elementary particles, physical fields, nuclei, atoms	Physical interactions
Chemical	Electrons, protons, neutrons, photons, atoms, molecules	«Pauli principle», which provides the order in which atomic orbits are filled and, consequently, the available set of atoms
Genetic	All living organisms	Genomes
Neuronal	Living organisms starting with worms	Neural systems
Communication	Collective (social) animals, humans	Signaling systems
A) Production-consumer	People	Communication systems, tools
B) Production-exchange	People	Communication Systems, tools and products of labor
C) production-financial- "...+..."	People	Technique, technology, social, financial and economic, government, international and global institutions, cognitive-communicative and network systems.

1.3. Basic (First) Law of Evolution and Complexity Theory

Let us now turn to the answer to the fundamental (in our opinion) questions of the theory of evolution formulated above. We argue that the cause and engine of all evolutionary processes without exception is the complexity of the evolving system. This means that an increase in the complexity of a system is a necessary condition for its evolution: if the system evolves, then its complexity increases. Thus, we come to an empirical generalization, which we call the First (basic) law of evolution and formulate as follows:

First Law of Evolution: The complexity of an evolving system always grows. In other words, an increase in the complexity of one or another mechanism of evolution is a *necessary condition for evolution*, in that part of it where this mechanism operates.

Of course, in order for the above statement to take on a real meaning, it is necessary to define the concept of complexity – the key concept of our theory. We adhere to and develop a phenomenological approach to the interpretation and application of complexity. In accordance with it, complexity is an exclusively multi-faceted, fundamental concept, comparable in its ontological and epistemological range with the concept of energy. However, they act differently and reflect different properties of things. So, if energy is a measure of *the intensity* of processes occurring in certain objects, then complexity is a measure of *heterogeneity, structure, irregularity of processes* and objects. Complexity, as well as energy, is characterized by an exceptional richness of forms of manifestation. If particular forms of energy are work, heat, electromagnetic, atomic, nuclear, gravitational energy, enthalpy, kinetic, potential, internal and even “dark” energy, etc., then the forms of manifestation of complexity are probability, fuzziness (in the sense of Zadeh), entropy, information (in the sense of Kolmogorov), fractality, etc. The noted facts are given in the table:

COMPARISON OF PROPERTIES OF ENERGY AND COMPLEXITY

ENERGY is a measure of intensity processes and objects	COMPLEXITY is a measure of the heterogeneity (structuredness, fuzziness, irregularity) of processes and objects
Dimension formula: ML^2t^{-2}	Dimension formula: $ML^2t^{-2}T^{-1}$
Units: $1 J = 0.239 cal = 1.602 \cdot 10^{-19} eV$	Units: Bit, $Ccal \cdot (K^{\circ})^{-1}$; K° – degrees Kelvin
Manifestations (forms) of energy In mechanics: kinetic, potential, ... Electromagnetic field energy: $\frac{E \cdot D}{2} + \frac{B \cdot H}{2}$	

In quantum mechanics: $E_{\psi} = \langle \psi H \psi \rangle$ (energy in the state ψ) ...energy and heat, energy and work, enthalpy, internal energy, free energy, energy and mass, dark energy... ,	Manifestations (forms) of complexity information, entropy, probability, fuzziness, «fractality»
--	---

It can be argued that energy and complexity are *complementary categories* (in the sense of *the complementarity principle of N. Bohr*), so that a sufficiently complete description of a system must contain both energy and complexity component. However, it should be noted that so far the role and influence of this latter has been underestimated. And even in theoretical works, complexity was considered mainly in the computational sense – as the complexity of algorithms and computer programs. However, already V. A. Uspensky, a student of A. N. Kolmogorov, spoke in the sense that if the 20th century was a century of energy, then the 21st century will already become a century of complexity. The author shares this point of view.

To operate with this concept, it is necessary to have some measures that allow estimating complexity from a quantitative point of view. The approach of A. N. Kolmogorov³ to the definition of algorithmic complexity can also help here, which, in our opinion, has great methodological strength and points the way to the development of a phenomenological approach in this area, which allows to quantify the complexity and its dynamics for objects and processes of the real world. Especially valuable in the mentioned work of A. N. Kolmogorov, in addition to the formal definition of complexity, is the direct relationship between complexity and information that he established, moreover, the direct expression of the amount of information of a constructive object through its algorithmic complexity. By projecting this formal approach onto reality, an insightful researcher will find conceptual and operational resources for determining – already at the phenomenological level – measures of the complexity of real objects and processes.

1.4. Measures and temporal dynamics of the complexity of the mechanisms of evolution. The Second law of evolution.

Note that the definition and construction of measures of complexity for mechanisms of evolution in real conditions and in certain phases of their dynamics is not an easy task and has not yet been solved. Approaches to it are currently semi-empirical in nature and are largely based on the intuition of the researcher. In our work [1], we presented complexity measures for various mechanisms of evolution and performed their comparison and analysis.

3 А.Н.Колмогоров. Три подхода к определению понятия количества информации. В кн. Теория информации и теория алгоритмов. – М.: Наука, 1987. – С.220.

In the table, as an example, we present the results taken from there for the genetic, neural and communication mechanisms.

OBJECTS OF THE EVOLUTIONARY SYSTEM	MECHANISMS OF EVOLUTION (COMPLEXITY PARAMETERS)		
	Genetic	Neural	Communication
	Volume of genomes (in bits)	Normalized volume of the neural system (in bits)	Volume of the sign («lexical») system (units)
Virus	10^3-10^4	0	0
Bacterium	10^5	0	0
Ciliates	10^6	0	0
Worms	10^6	10^0-10^1	0
Bees, Ants	10^7	10^3-10^4	100
Chimpanzee, Dolphins	10^8	10^7-10^8	10000
Human	10^9	$10^{10}-10^{11}$	From 10^7-10^9 to $>...$

The table shows that the complexity parameters for the genetic, neural and communication mechanisms are, respectively: the volume of genomes, the normalized volume of neuronal systems and the volume of "lexical" systems. It is also seen that over time, as new species appear, an increase in each parameter is observed. At the same time, each of them arises "at its own time", namely, when its predecessor has already, as it were, "exhausted" the intra-system possibilities of its growth. Another conclusion concerns the acceleration of the growth rate of complexity parameters. The analysis shows that if in the early stages of genetic evolution the doubling time of the complexity parameter (genome volume) was more than 60 million years, then the doubling of the communication parameter for humans, judging by the studies of S.P. Kapitsa in the field of human population growth⁴, occurred in a progressive pace (starting from one thousand years in the era of the Neolithic revolution and up to 50 years by the beginning of the twentieth century. That is, the growth rate increased by about 20 times in 10 thousand years). From what has been said, the following conclusions can be drawn.

1. All mechanisms of evolution have one thing in common: the unfolding in time of any of them leads to the complication of the evolving system. This means that in the course of the temporal dynamics of a specific mecha-

nism of evolution, its complexity (estimated by the value of the corresponding quantitative correlate) increases.

2. At the same time, each evolutionary mechanism – the mechanism of evolution through the complication of the system – has its own natural limit. It is, as it were, "programmed" to achieve a certain maximum complexity characteristic of this particular mechanism.

3. If an evolving system approaches this limit (its complication resources are approaching exhaustion), it falls into the so-called "complexity trap", enters a critical phase and "trying to find" other mechanisms that can ensure its complication, and therefore continue the process evolution.

4. In the case under consideration, another one was connected to the genetic mechanism (when it approached its limit) at a certain stage – the neural mechanism and evolution was continued. Then the neural mechanism came close to exhaustion and the communication mechanism kicked in.

Let's summarize what has been said. As far as we can judge from the analysis of the available data, the temporal dynamics of all mechanisms of evolution have common features. They are shown schematically in Figure 1.

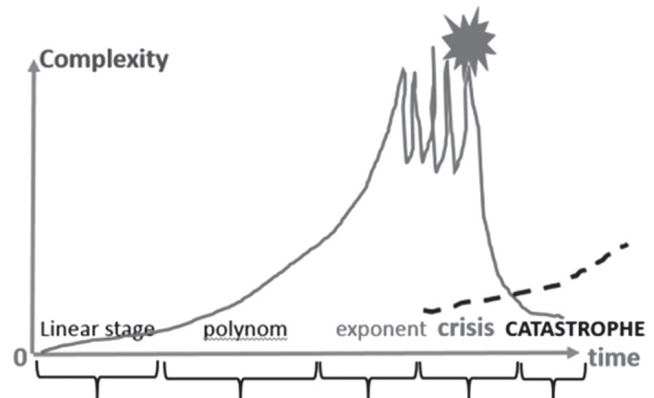


Fig. 1. Generalized dynamics of a mechanism of evolution

Thus, at the initial stage of deployment of the evolution mechanism, the complexity grows linearly in time. This era is relatively calm, predictable development of an evolving system, its "golden age". During this period, the formation of the main system characteristics and determinants of the system and their exit to the stage of dominance takes place.

Then comes the era of polynomial growth in complexity. The system develops its characteristic properties. At the same time, development accelerates, revealing all the potential possibilities of the corresponding mechanism of evolution.

After that, the system enters a phase of exponential growth in complexity. A characteristic feature of the exponential period is that it occurs at the border of the limiting possibilities for the development of the system within the framework of the corresponding mechanism of evolution. This period of complication is poorly amenable to or is no longer amenable to control and management, and critical phenomena begin to grow in the system. Their reason

4 Капица С.П. Феноменологическая теория роста населения Земли. // УФН 166 63 (1996) ; Капица С.П. Общая теория роста человечества: сколько людей жило, живет и будет жить на Земле (М.: Наука, 1999).; Капица С.П. Парадоксы роста. Законы развития человечества. М.: Альпина нон-фикшн, 2009. ; Kapitsa S P Global Population Blow-Up and After. The Demographic Revolution and Information Society (A Report to the Club of Rome and the Global Marshall Plan Initiative) (Hamburg: Tolleranza, 2006, Moscow, 2007).; С.П.Капица. К теории роста населения Земли. УФН, т. 180, № 12, 2010 Сс.1337-1346.

is the internal inconsistency of the evolutionary process: The basic law of evolution requires an increase in complexity, and the need to ensure the controllability of the system with the same necessity limits this growth to an exponential region. Critical phenomena are growing and can become catastrophic for the system. At this time, in the depths of the system (and, possibly, outside of them) there occurs (if it occurs) the birth of a new mechanism of evolution (in the figure it is indicated by a dashed line), with which approximately the same story occurs.

The foregoing is the content of the **Second Law of Evolution** – the Law of the temporal dynamics of the mechanism of evolution.

1.5. Parasitic complexity phenomenon. The Third law of evolution.

From the foregoing, one might get the impression that we imagine evolution as a monotonous process of a steady increase in the complexity of the mechanism of evolution until it enters the critical region. Such a conclusion could be drawn if the “Generalized dynamics of the mechanism of evolution” depicted in Fig. 1 is taken too literally. 1. But in fact, this is not the case, and the picture looks much more ambiguous.

First, reality is arranged in such a way that it seems to “resist” the evolutionary process, which is accompanied by an increase in the complexity of the system. That is, in parallel with the evolutionary processes, anti-evolutionary processes can and do go on in the system. These effects can be traced in every mechanism of evolution.

The following observation refers to the phenomenon of anomalous growth in complexity observed in almost all mechanisms of evolution. The noted anomalous growth can be summed up in one term “inflation”.

For example, in the genetic mechanism, as in any other, the phenomenon of the “complexity trap” operates. One of its manifestations, in our opinion, is the existence of genomes of organisms that are simpler in their general organization than higher animals and humans, but whose genome complexity exceeds that of the human one. For example, the wheat genome consists of approximately 17 billion base pairs, which is approximately 2.5 times higher than the corresponding indicator of the human genome (6.9 billion base pairs). Some species of amoeba generally differ in the colossal length of the genome. Thus, the genome of the amoeba *Amoeba dubia* consists of 690 billion base pairs. There is a large bibliography on this subject. It is partially presented in our work [1]; here we will give only two, in our opinion, the most characteristic references to two remarkable works⁵. However, even they are far from exhausting the subject.

5 Щербаков В.П. Эволюция как сопротивление энтропии// Журнал общей биологии. 2005. 66(3) : 195–211; А. В. Марков, В. А. Анисимов, А. В. Коротаев. Взаимосвязь размера генома и сложности организма в эволюционном ряду от прокариот к млекопитающим // Палеонтологический журнал. №4, 2010. С. 3–14.

The situation is similar with the financial mechanism. Following [1], we take the global volume of GDP as a parameter of the complexity of this mechanism and present the corresponding data from this work. So, for the period 1970–2016. World GDP increased from \$3398.7 billion to \$75212.7 billion and increased by \$71817 billion or 22.13 times. A 22-fold increase in 45 years cannot but look impressive (if not frightening). In more detail financial evolution will be considered in the second part.

Something similar happens in the field of the physical mechanism. In connection with the discovery of the anomalous, time-accelerating expansion of the universe, scientists eventually came up with the so-called lambda-CDM model. This model assumes that in the early stages of the evolution of the Universe (age from 10^{-43} sec to 10^{-37} sec), it underwent a colossal expansion, which led to exponential growth of all spatial scales. This phenomenon is called the “inflationary model”, since the term “inflation” in cosmology means a rapid growth in scale, in which the growth rate is proportional to the value of the scale itself. However, based on the observations of supernovae carried out in the late 1990s, it was concluded that the expansion of the Universe is accelerating with time, and such expansion refers to the present, and not to the early stage of the Universe that took place immediately after the Big Bang. These observations were supported other sources (measurements of relic radiation, gravitational lensing, Big Bang nucleosynthesis). As a result, the analysis of the phenomenon of the accelerating expansion of the Universe led to the need to introduce the concept of “dark energy” responsible for such an anomalous expansion.

Recently, Roger Penrose proposed a new interpretation of this behavior of the Universe, different from the lambda-CDM model. In his remarkable lecture “Seeing Trough the Big Bang into Another World” (“Is it possible to see the previous Universe through the Big Bang?”), he suggested that the early (“post-Planckian”) inflation of the Universe is nothing more than a trace of its previous (!) evolution, which led to the catastrophe (!) of the Big Bang, and the late, modern inflation, accompanied by exponential expansion, is already a consequence of its modern natural evolution.

The author is very impressed with R. Penrose’s model. It fully corresponds to our evolutionary concept. Moreover, it draws attention to the need to analyze the traces of the previous evolution, which led to the catastrophe of the previous World and the emergence of a New One. In addition, it suggests that the Big Bang, which laid the foundation for our Universe, had been possessed with the evolutionary nature identical to the mechanism of our Second Law of Evolution – the law of temporal dynamics. Finally, the above methodology leads to conclusions that clarify the supreme meaning of “dark” energy as the evolutionary factor and so the need for its appearance. Of course, this ideology – with appropriate clarifications

– is applicable not only to the physical, but also to any other mechanism of evolution. This issue will be analyzed in detail in the second part of our work.

The above example and effects from completely different areas, it seems to us, are associated with one property common to all evolving systems, which we called the “parasitic complexity phenomenon”, which begins to manifest itself especially clearly and even dominate at a certain, rather late stage in the unfolding of any mechanism of evolution.

The fact is that **not every increase in the complexity of a system leads to its evolution**. Under certain circumstances, the system gets the opportunity to “accelerate” its complexity to unthinkable values, while practically not evolving, but in the process driving itself into a critical region. We have to state that the source and cause of evolution – the increase in the complexity of an evolving system, sooner or later begins to give rise to crises or even catastrophes in it. As noted above, increasing complexity is just a **necessary condition for evolution**. But, unfortunately, it is not **sufficient**.

Above, we have given examples of a colossal (exponential, “explosive”) growth in the complexity of the mechanisms of evolution with an unimpressive evolution of the corresponding systems. As a result, a situation arises of an uncontrolled, “parasitic” increase in the complexity of the system, which is not directly related to its evolution. This circumstance gives us reason to single out the parasitic component in the complexity of the system and present the total complexity of any evolving system as a sum:

$$C_{total} = C_{evol} + C_{parasite}, \quad (1)$$

where C_{evol} is the part of complexity corresponding to real evolutionary processes; $C_{parasite}$ – parasitic complexity not related to evolution; C_{total} is the total complexity of the system. We introduce the value σ (σ -factor) as a factor in

the efficiency of the evolution of the system, according to the formula:

$$\sigma = \frac{C_{evol}}{C_{total}} = \frac{C_{evol}}{C_{evol} + C_{parasite}}; \quad 0 \leq \sigma \leq 1. \quad (2)$$

Obviously, the σ -factor reaches its maximum value of 1 only in the ideal case, when $C_{parasite} = 0$.

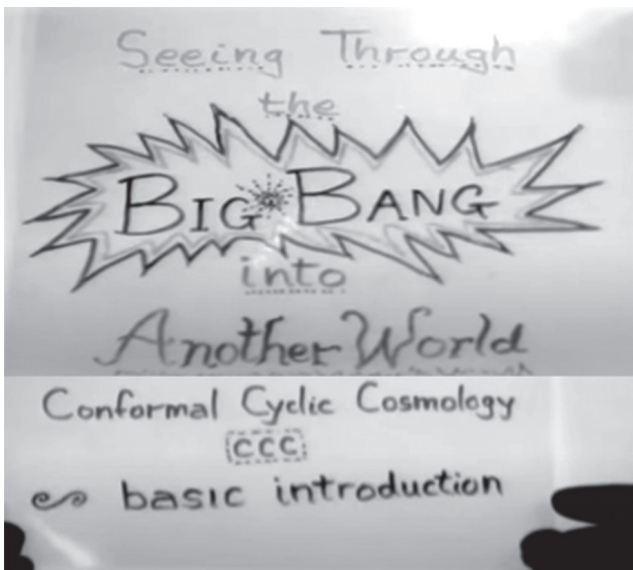
The foregoing gives grounds for the formulation of the **Third Law of Evolution – the law of increasing “parasitic” complexity**, the brief essence of which is set out in the two previous paragraphs.

Part II. LIGHT, GRAY AND DARK SIDES OF BEING

2.1. What is wrong with our financial system?

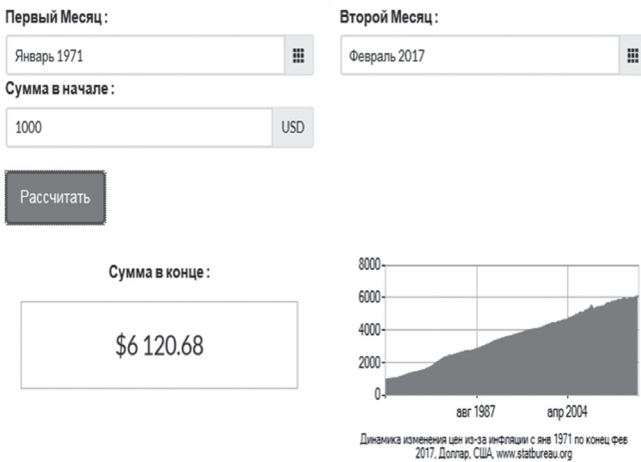
A striking example of the operation of the law of increasing “parasitic” complexity is the behavior of the global financial system over the past few decades. Let’s try to estimate the value of the σ -factor for its main, dominant subdivision, namely, the one that is based on the dollar as the main world reserve currency with its principal and, at the same time, factotum – the US Federal Reserve System.

We will proceed from the assumption that in the financial system the main contribution to the production of $C_{parasite}$ is made by inflation. Of course, in this context, the term “inflation” is used not in the cosmological, but in the financial sense. However, in fact, in this situation, these meanings paradoxically coincide, which leads to the idea of their isomorphism, and that behind this is the same general pattern. Namely, from the law of temporal dynamics and the law of “parasitic” complexity it absolutely unequivocally follows: the complexity of an evolving system will grow in time, and at an ever accelerating pace, however, the evolutionary value of this complexity will decrease in the same proportion, acquiring an increasingly “inflationary” character.



Roger Penrose lectures: «Seeing Trough the Big Bang into Another World»
 (<https://www.youtube.com/watch?v=5IB9m49zUag>)

To estimate the value of the “parasitic” term, we will use the program posted on the website⁶, which allows you to calculate the inflation rate of the US dollar for a certain period online. Let’s calculate this level, starting from 1971 to 2017:



Calculation of dollar’s inflation dynamics for the period “January 1971 – February 2017”

We calculate the efficiency value σ as an “inflationary” ratio:

$$\sigma = \frac{1000}{6120.68} \approx 0.16, \quad (3)$$

where 1000 is the amount at the beginning (January 1971), and 6120.68 is the amount at the end (February 2017), taking into account dollar inflation for this period. This calculation shows that at least **83% of the resources of the financial system during this period were spent on the production of “parasite” complexity $C_{parasite}$** . It is possible that the real value of the σ -factor is actually much smaller, since we took into account only one “parasitic” parameter – the inflationary source of $C_{parasite}$ growth. But even from such a rough estimate, it can be seen that the **“efficiency” of the modern global financial system is extremely low, and if this evolutionary trend continues, it will progressively decrease.**

Of course, there are many questions here. What are the visible and invisible causes of “parasitic” complexity? To what extent is it possible to deal with them in principle? How to increase the σ -factor and decrease $C_{parasite}$? To what extent is it generally possible (if possible) to improve the efficiency of the existing financial system? Who and what does it depend on? No less important is the question: who mainly bears the burden of “maintenance” of “parasitic” complexity and vice versa – who is the main beneficiary of this status quo?

In order to answer these questions, it is necessary, at least very briefly, to dwell on the history of the financial mechanism and its substance – money. In this presentation, we will follow our work [1].

2.2. Brief history of the modern World Financial System.

And what do the laws of evolution have to do with it?

According to history, money appeared about 5 thousand years ago. Around the same time, the first state formations appeared. Since then, the history of mankind has been associated with the history of money. This history is boundless and a colossal and also boundless literature on economics, politics, technology, technology, history, geography, organization, culture, ethics, aesthetics, morality, psychology, medicine, law, forensics, etc., is devoted to it of money. This entire period was associated with the monopoly of the state on the issue of money, management and control of financial processes. However, in 1913 this monopoly was abolished in the United States, where the so-called Federal Reserve System (also known as the **Federal Reserve** or simply **the Fed**) was created, which acts as the Central Bank of the United States, but is a private organization. It was with the creation of the Federal Reserve in the United States that the state’s monopoly on the issuance of money in favor of the Fed was eliminated, which “pulled over” the role of a monopolist in this matter.

The subsequent events (World War I, Great Depression, World War II, the creation of the Bretton Woods system, etc.) suggest some kind of teleology in the creation of the Fed, aimed at constructing the World Financial System in the future – a kind of global financial pyramid, the top of which is the Fed.

The author is far from conspiracy theories – rather, the ideologists, creators and factotums of the Federal Reserve managed to take full advantage of the opportunities provided to them by the vicissitudes of historical development. However, one cannot deny the fact that such a long-term and purposeful political and financial stratagem could hardly be realized if it had not been induced and stimulated by certain objective laws. We refer to these as the Basic Law of Evolution, because the development of the World Financial System was accompanied by a steady increase in the complexity of the corresponding mechanism of evolution, which will be demonstrated below.

It seems to us that the leaders of the Fed step by step realized the new opportunities provided by the private status of a monopoly-producer of the currency of a country like the United States. Already after the First World War, where the United States, without making any special efforts, managed to take advantage of a significant number of its results, it became clear that in the new world, much more closely connected by various kinds of political, economic and other relations, the role of money as a universal agent intermediary between producers and consumers, subjects and objects of the global social scene will increase. And the winner will be the one who will be able to adapt to the new conditions to a greater extent.

In 1933, US President Franklin Roosevelt signed Executive Order No. 6102 on the actual confiscation

⁶ <https://www.statbureau.org/ru/united-states/inflation-calculators?dateBack=1971-1-1&dateTo=2017-2-1&amount=1000>

from the population and organizations of gold located in bars and coins. Thus, the gold standard, if not completely eliminated, was significantly limited. In fact, this was not just a statement of the fact that, due to the greatly increased scale of the economy, gold is no longer able to cope with the function of a money stabilizer, but mainly an action aimed at providing new degrees of freedom in the financial sector to perform various kinds of maneuvers, including speculative. The Great Depression in the United States served as an excuse (or maybe it was the reason?) for the restructuring of the economy under the tasks of dominating financial and political institutions over purely industrial ones.

An exceptional event in terms of the prospects for expanding the influence of the dollar system was the Second World War. One gets the impression that it seemed to have been specially organized in order to ensure the dominance of the dollar in the world. To paraphrase a well-known expression, we can say that if the Second World War had not broken out, then it would have to be invented only so that the dollar would take a dominant position in the world⁷.

Even before the end of World War II, in July 1944, the so-called The United Nations Monetary and Financial Conference, better known as the Bretton Woods Conference, took place in Bretton Woods (USA, New Hampshire). It was intended to lay the foundations for the post-war financial world order.

In particular, the Bretton Woods system was established – historically the third in a row, according to which a fixed price was set for gold (35 dollars per 1 troy ounce). As a result, the United States, represented by the Fed, gained world currency hegemony. In fact, this led to the emergence of the Gold Dollar Standard of an international monetary system based on the leadership of the dollar. In the middle of the 20th century, the United States owned 70% of the world's gold reserves. The dollar has become the base of currency parities, the predominant means of international settlements, foreign exchange interventions and reserve assets. The national currency of the United States, and in fact – the currency of the Federal Reserve became at the same time world money.

It is characteristic that the USSR, despite the fact that it was a participant in the conference and signed the Bretton Woods agreements, did not ratify them in the future, which severely limited the integrity of the world financial system and, therefore, the hegemony of the dollar. In our opinion, this became one of the main reasons for the beginning of the Cold War, since it objectively narrowed the area of the World Financial System based on the dollar as the main reserve currency.

In 1976, at the conference of the IMF countries in Kingston (Jamaica), a new – the fourth world monetary

system was approved. It attempted to eliminate some of the contradictions of the Bretton Woods system. The gold standard was abolished and the demonetization of gold was legally completed: its official price, gold parities were canceled, the exchange of dollars for gold was stopped. Now purely financial substances are almost completely freed from the restraining bonds of matter in the form of gold, which further stimulated the activation of their "non-material", "subjective", "informational" properties. The role of the SDR was declared as the main financial instrument of the IMF⁸, but the dollar still remained the leading world currency. Thus, a highly effective symbiosis of the Fed system and the US state was finally formed. The national idea and national treasure of the United States – the dollar continued its victorious campaign in the World, since now almost all the necessary institutions have been created to form the global World Financial System. The dollar, which became the world currency, more and more definitely won the place of the main substance and subject of the financial mechanism of evolution. The "financial postmodern" has begun.

The consequence of this status quo was the possibility of using dollar-based global financial instruments to solve the US's own problems. The reverse side of this opportunity was the conviction of the United States in its exclusive role in the world. This role, which manifests itself in completely different forms (and in our time even in those that resemble *idée fixe*), in fact, is based on the exclusive financial status of the Fed – the status of the principal of the global financial dollar pyramid. At the same time, it is necessary to realize that this pyramidal structure crowned by the Federal Reserve, performing the functions of an evolutionary factor, can exist only in conditions of constant growth of its distribution area.

From some detached point of view, this process looks identical to the modern inflationary-exponential expansion of the Universe, which we mentioned above. Therefore, the processes of globalization of the world that began in that era actually demonstrated nothing more than a natural form of evolution of the World Financial System itself, subject to our First, Second and Third Laws. So the laws of evolution really have something to do with it. And the historical role of the United States and its Federal Reserve was reduced to supporting and accompanying all the tools and institutions that ensure this evolution.

The foregoing formed the basis of the new World Order. This expansion was limited only by the existence of a socialist system associated with the USSR, which was not part of the dollar pyramid, which became the main reason for its dismantling, which ended with the liquidation of the USSR in 1991. Thus, all obstacles to globalization, and in our terminology – unrestricted «inflation» – were swept away. "To develop", in the sense of fighting with

⁷ The author reminds that he is not based on conspiracy theories. Our conclusions are invariant with respect to the facts of the existence or non-existence of a «world conspiracy» or a «secret government»

⁸ <https://www.imf.org/en/About/Factsheets/Sheets/2016/08/01/14/51/Special-Drawing-Right-SDR>

restrictions, was no longer with anyone, because there were no longer any borders. This gave Francis Fukuyama reason to imagine that history was already over⁹.

But in fact, it has only just begun.

After all, the next decade has convincingly demonstrated that in reality we are dealing not so much with the end of history as with the beginning of a certain new stage in it. Which confirms the well-known idea that history is still ahead of thinking. For us, this new stage is associated with the formation of the financial-information-network mechanism of evolution.

2.3. Financial-information-network (FIN) mechanism and assessment of its complexity

We emphasize that the onset of certain stages of the Production Mechanism occurs, as they say, “in due time” and correlates with the corresponding social and technological phenomena of an innovative nature. In this regard, we note the chronology of the latest stage of financial, as well as informational and network evolution.

We associate the beginning of the newest historical stage of the financial phase with the creation of the Fed in 1913, after which a number of social upheavals of a global nature occurred (World War I, the collapse of the Russian Empire and the creation of the USSR, the Great Depression in the USA, World War II, the creation of the Bretton Woods system, the collapse of the British Empire). We attribute the beginning of the information phase of the FIN stage to 1974 and associate it with the first world energy crisis. A detailed justification for such an accurate dating is given in our work [1]. The beginning of the network phase of the FIN stage – around 1990 – we associate with the creation and spread of the Internet. We attribute the unconditional approval of globalization as the main trend of world development to 1991 and mark it with the liquidation of the USSR.

It is necessary to note the content heterogeneity of the factors FIN-stage, namely: the Internet is an information technology factor, the collapse of the socialist system is political, and globalization is socio-economic. All of them contribute to determining the measure of the complexity of the evolutionary mechanism and its dynamics.

Above, we analyzed the financial aspects of complexity, which concentrate the socio-economic and political components.

In recent decades, information and network aspects have become especially active. At the same time, in

scientific papers devoted to the theory of complex networks, their complexity, as a rule, is not analyzed quantitatively.

At best, a reduction to some computational scheme is proposed, for which the computational complexity can be estimated, which is then compared to the complexity of the analyzed system. It is clear that in a literal setting to use such approaches in our case is not entirely appropriate. Based on the phenomenological understanding of complexity, we choose the total amount of information circulating in the Web as an integral measure of the complexity of the FIN-stage of evolution.

This indicator has recently been actively scanned. In particular, according to IDC, the amount of data stored on the Internet approached 487 exabytes (487 billion GB) in 2009 and was projected to double within about a year and a half. But already in 2012, IDC announced that in 2011 the total global volume of data created and replicated by humanity exceeded 1.8 zettabytes (1.8 trillion GB), more than double the previous forecast rate. According to further IDC forecasts, the amount of data on the planet will at least double every two years right up to 2020, which has happened and continues to happen in our time.

There are two important aspects to this process. The first concerns the actual volumes of information that humanity is no longer able to process and comprehend. The second is related to the fact that, according to some estimates, most of the information in circulation is actually parasitic in nature and does not contribute to the evolution of mankind. In these aspects, the laws of evolution find their manifestation, prescribing as a necessary condition an increase in the complexity of an evolving system, and as the volume and rate of growth of complexity increase, an uncontrolled growth of its parasitic component, which does not contribute to the evolution of the system, but constitutes an unbearable load on its functional subsystems. Thus, **the global information system of mankind, along with others, has now acquired a crisis, almost catastrophic character.** And indeed, due to the network, inflationary nature of modern civilization, these **critical phenomena are characterized by universality, they are broadcast to all the main aspects of its functioning and generally threaten the existing World Order.** This is manifested in the permanent crises of the global financial system, international, interstate and humanitarian relations, etc.

So, if we rely on the data growth rate of information on the Web as a quantitative correlate of the measure of complexity of the FIN-stage of evolution, then we should conclude that **the world has entered a critical – exponential period of growth in the complexity of its FIN of the evolutionary mechanism.**

As for crises, there is much more data here. For example, in the financial sector. As noted in [1], in the period from 1970 to 2013, there were more than 400 major

⁹ As early as 1989, Francis Fukuyama’s essay «The End of History?» appeared in The National Interest. (Francis Fukuyama, «The End of History?»), and in 1992, based on this article, his book saw the light of day under the eloquent title «The End of History or the Last Man» (Francis Fukuyama. The End of History and the Last Man. 1992. Free Press, P. 418;), without the question mark in the title. In this regard, the author cannot but recall the ironic expression of a much more insightful writer – M.E. Satykov-Shchedrin from his «History of a City»: «History has stopped its course.»

banking, currency or debt crises. Internet data on this subject go off scale¹⁰.

Critical phenomena have swept over the Network itself. Recently, global hacker attacks have been extremely intensified, so they are already talking about full-scale information wars, cyber wars and hybrid wars. Cyber troops are being created in the military and special structures of various countries.

We will return to the analysis of critical phenomena, since their common internal source, as it seems to us, is the uncontrolled increase in the complexity of the corresponding systems and the resulting impossibility of their effective management. In fact, in our time, global governance has been reduced to one parameter – the volume of the dollar issue, and elementary control theory affirms the fatal inevitability of crises in a system of great complexity, controlled by just one parameter.

Thus, we come to a statement of the **main contradiction of the modern era**, which we formulate as follows:

“On the one hand, there is a **need to increase the complexity of the world system to ensure its evolution**, and on the other hand, and **at the same time, there is a need to reduce the same complexity** due to the impossibility of effectively managing a system of such great (exponentially growing) complexity that the “Network of Networks” acquired.

Analysis of the main contradiction thus formulated allows us to write down the symbolic relationship between the controllability of the system and its complexity:

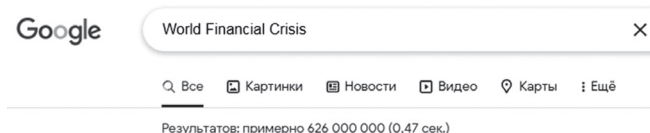
$$\langle \text{Controllability} \rangle \cong \frac{1}{\langle \text{Complexity} \rangle}, \quad (4)$$

Formula (4) demonstrates that the controllability of a system is inversely proportional to its complexity; that is, the more complex the system, the more difficult it is to manage. In the limiting case (at the singularity point) $\langle \text{Complexity} \rangle \rightarrow \infty$, therefore, $\langle \text{Controllability} \rangle \rightarrow 0$. Thus, when entering the exponential zone of complexity growth, the system falls into the “**exponential uncontrollability hole**”, that is, it becomes practically uncontrollable, which serves the main cause of the crisis phenomena developing in it, which can even lead to a catastrophe and the collapse of the system.

2.4. System structure of an evolving object. Consequences of growth of parasitic complexity.

The stated facts and the conclusions following from them regarding the control of evolving objects located in the exponential well are depressing. Obviously, the simultaneous satisfaction of such defiantly opposite requirements as those formulated above, basically a

10 As of October 13, 2022:



contradiction, is impossible. But there is no hope that the problem will “solve itself” either. Therefore, it is necessary to propose some measures. For example, temporarily sacrifice evolution in favor of the controllability of the world system. Or take a risk and surrender to the will of the evolutionary elements. Or vice versa – to more carefully analyze the sources of parasitic complexity, localize them and take vigorous measures to eliminate them or at least reduce their influence. Or something else. To begin with, at least analyze the considerations we have outlined about the causes of modern crises.

However, the author is quite conscious of the fact that the world bodies exercising global governance (if any) are not at all obliged to know our theory. At the same time, an analysis of world processes allows us to conclude that the mentioned bodies, apparently, are guided by ideas, if not literally coinciding with ours, then, in essence, very close to them – if not in form, then in anyway, in terms of content. Moreover, it seems that they even know something more, with which we congratulate them. We are led to this conclusion by the options for overcoming global crises. In [1], we considered and analyzed several such options. Let’s list them.

Option number 1. Controlled chaos.

Option number 2. Modification and transformation of the MFS (doctrine of “controlled complexity”).

Option number 3. Restriction of the integrity of the MFS.

Option number 4. Cosmic evolution.

Option number 5. Scientific and technological progress.

Option number 6. Quantum information and quantum civilization.

Due to space constraints, we are not in a position to analyze all these options in detail here. Therefore, we refer the reader to our work [1], and here we will say a few words only about option No. 1 – controlled chaos.

We have to admit that it is the most practically developed and even actively used in practice in our time. His main method is based on the empirical observation that if instabilities of not very clear origin arise in a complex system, then a quasi-random “chaotic” effect is introduced into the system to combat them. Experience shows that during such an operation, the system very often “calms down” for a while and its parameters stabilize. The doctrine of “controlled chaos” was developed in the USA. Its authors usually include Zbigniew Brzezinski (author of The Grand Chessboard: America’s Dominance and Its Geostrategic Imperatives), Gene Sharp (author of From Dictatorship to Democracy), and Steven Mann, who published in 1992 in the National War College Journal in Washington, D.C. “Chaos Theory and Strategic Thought”. All of them are related to the “color revolutions” in the

countries of the former USSR, Yugoslavia, and then in Egypt, Tunisia, Libya, Syria, as well as in the post-Soviet countries (Ukraine, Georgia, Kyrgyzstan, ...). Numerous examples of the application of the “controlled chaos” method can be observed in our time..

Unfortunately, it should be recognized that “controlled chaos” is apparently not a good solution to the problem, and the stabilization of the world order is getting worse and worse, since its means and methods are themselves very poorly controlled and therefore fraught with unpredictable consequences up to a global catastrophe, which make all “measures taken” meaningless. Especially dangerous here is that this catastrophe can happen quite by accident, as a result of the action of the notorious “human factor”.

As for the other options, theoretically, as far as we know, they are being developed. However, the real and tangible practical results of this activity are unknown to us.

2.5. Crisis of “Corona” and its evolutionary-systemic interpretation.

As it was said at the beginning of our article, in late 2019 – early 2020, a global pandemic caused by coronavirus SARS-CoV2-COVID19 arose and began to develop. The first outbreak was recorded in Wuhan, China in December 2019. Already on January 30, 2020, the World Health Organization declared this outbreak a public health emergency of international concern, and on March 11, a pandemic. Thus began the global social crisis. It was aggravated by the fact that the population in its mass, and, apparently, most governments had little or no idea of the causes, threats of this disaster and effective methods of dealing with it. As a result, panicky rumors spread, outbreaks of social destabilization spontaneously arose, and so on.

In fact, as of mid-March 2021, over 120 million cases of the disease have been registered worldwide; more than 2.6 million people have died and about 91.2 million have recovered, and COVID-19 has begun to decline. True, new strains of it began to appear along the way, new outbreaks were observed in different regions, but they no longer had a global character.

However, the COVID-19 pandemic has had severe social and economic impacts, including the largest since the Great Depression global recession and massive famine affecting some 265 million people.

This has led to the postponement or cancellation of numerous sports, religious, political and cultural events, and widespread supply shortages have been exacerbated by panic buying.

Schools, universities and colleges were closed either on a national or local basis in 172 countries, affecting approximately 98.5% of the world’s school and student populations.

Numerous sporting events have been canceled or postponed, including European national championships in

almost every sport; The European Football Championship 2020 has been postponed for a year, as well as the 2020 Summer Olympics, which were supposed to start on July 24 in Tokyo.

Numerous cultural events such as festivals, exhibitions and competitions have been canceled or postponed, including the Moscow and Cannes International Film Festivals, the Farnborough Air Show, the Eurovision Song Contest 2020, etc.

The COVID-19 pandemic – we call it the Corona Crisis – has dealt a severe blow to the economies of all countries. According to experts, the most serious economic crisis since the Second World War was expected.

The authorities in some countries have tried to provide massive financial support to companies, families and the most vulnerable groups. In total, since March 2020, according to some estimates, about \$12 trillion has been spent on these purposes.

However, the infusion of public money did not save the economy from falling, and sometimes collapse due to the cessation of business activities, travel, disruption of supply chains, and so on. The fall in global GDP, according to the Organization for Economic Cooperation and Development (OECD), was expected at 4.5%.

A report from the United Nations Conference on Trade and Development (UNCTAD) released at the end of November 2021 warned that the crisis could push another 130 million people out of poverty and worsen inequality and social protection.

In the first nine months of 2020, global hours of work fell by 11.7 percent, equivalent to the loss of 323 million permanent jobs, according to the United Nations International Labor Organization (ILO). The reduction in working hours resulted in \$3.5 trillion in revenue loss, or 5.5% of global GDP.

It seemed that the Corona Crisis brought the world to the threshold of an unprecedented catastrophe, but to everyone’s surprise and massive eschatological expectations, nothing of the kind happened.

At the end of March 2020, the Chinese media called the coronavirus pandemic the beginning of the de-dollarization of the world. The Chinese edition of The 21st Century Business Herald published an article in which the journalist outlined the economic problems of the United States during the pandemic, and also explained why the COVID-19 crisis could take away the status of the world reserve currency from the dollar, that is, lead to de-dollarization of the world. Over the years, China has assumed an increasingly important role in trade relations, but the dollar has only strengthened its position as the dominant reserve currency. Despite the Belt and Road Initiative, which was supposed to promote a new model of international cooperation and development, China’s share of the trade sector, denominated in yuan, was only 14%. Since the 2008 crisis, the volume of non-bank debt in US dollars

almost doubled, while the share of other major currencies stagnated or declined. The Corona crisis, as many hoped, would intensify de-dollarization processes. But that didn't happen either.

The question arises: why?

It seems to us that the Corona Crisis, despite its humanitarian and dramatic nature, was used by the ruling circles in the way it should have been used – in a completely pragmatic vein, namely as an element of “controlled chaos”. In what follows, we will use such an interpretation of “controlled chaos” that is consistent with our evolutionary approach.

There is no doubt that the Corona Crisis introduced a disturbance into the world socio-economic system, and, unlike, so to speak, the standard disturbances of “controlled chaos”, it no longer had a local, but quite a global character. At the same time, it remained manageable to an acceptable extent, since the actions of governments, one way or another, “on average” coincided with the desires and aspirations of the peoples and were aimed at protecting against a danger that was seen as quite mortal. As usual, the fear of death was the most effective factor.

Consciously or due to circumstances – it does not matter, but the measures taken in their mass contributed to the reduction of the “parasitic” complexity of the global economy and finance. Let's list these factors:

1. Decrease in activity in the field of show business, sports, tourism, other aspects of the “consumer society”.
2. The growth of unemployment, the reduction in the total amount of working time.
3. A sharp increase in distance, online methods of work and education.
4. Hunger.
5. Ban on mass gatherings.
6. Lockdowns.
7. A sharp decline in the activity of stock markets.
8. Reducing the environmental burden.

No matter how cynical and inhumane it may seem at first glance, but an elementary analysis shows that in each of the above factors one can feel and designate in words a “antiparasitic” component. Thus, the World Financial System with the dollar at the head has withstood and, as it seems to us, even somewhat strengthened. And this despite the seemingly completely inflationary multi-trillion dollar emission during the pandemic¹¹, designed to compensate for business losses and stimulate business activity.

Despite the intuitively felt “antiparasitic” nature of the factors listed above, it is not possible to more or less correctly quantify their effect. Anyway, the author does not know how to do it. We have to state only the most general features of what happened, namely:

*“CROWN” showed the characteristic evolutionary features of society: **on the one hand**, the presence of “official”, “permitted”, “legal”, “legal”, “permitted”, etc. etc. , and **on the other** – not quite permitted, and even directly prohibited forms of social structure, hostel and functioning..*

2.6. Fundamental structure of an arbitrary rather complex system

Let's move on to generalizations.

In connection with the foregoing, there is a strong feeling that there are some hidden parameters operating in the world in general. They seem to be affected sometimes, but not always. This requires some special approach and special tools with which you can manage situations within certain limits. Let us try to carry out a small analysis of the controllability of systems, if possible, without resorting to standard methods of control theory.

Controllability (*manageability*), first of all, implies that the subject of management has tools to control the parameters of the managed object. For this, of course, it is necessary that they be known. Then it is necessary to find out which of these parameters are observable quantities and which are not. For the classical assumption of absolute observability has long been refuted. The idea of observability, which was developed in detail for the first time in the history of science when substantiating quantum mechanics, is, on the whole, very productive from a methodological point of view as applied to the description of complex processes. As A. Einstein noted in his discussion with W. Heisenberg in 1926 during a discussion of the foundations of quantum mechanics¹², it is impossible to build a theory only from observable quantities (although without them, no scientific theory is generally unthinkable). Finally, only the theory itself should determine which of its quantities are directly observable and which are not. In particular, in quantum mechanics, states are, generally speaking, not directly observable quantities. The latter correspond to operators of a certain type, acting in the state space and whose eigenvalues correspond to the values of observables.

Thus, in the world, and hence in cognition, there is an “invisible”, that is, something that cannot be directly observed. This gives us grounds for recognizing the role of faith in cognitive-intellectual processes, which, in the words of the Apostle Paul, is “certainty in the invisible”¹³.

However, it should be recognized that the deeper epistemological reasons for observability are not disclosed in quantum theory either. We believe that these reasons lie in the phenomenology of the complexity of manifested being. Any reality, being an object of observation (in other words, entering into a “subject-object” relationship),

11 According to some reports, the amount of money supply “under COVID-19” in the United States amounted to 5.9 trillion dollars, which, in turn, is 38% of the total money supply, and is also equal to the volume of emission over the past 40 years.

12 Гейзенберг В. Физика и философия. Часть и целое. Пер. с нем. – М.: Наука, 1989. – Сс. 191-196.

13 “Faith is the substance of things hoped for and the certainty of things not seen.” St. Ap. Paul, Heb 11:1.

manifests its essence in some limited forms that can be perceived by the subject. The manifested complexity of the noted forms (potentially it is infinite – after all, matter has no boundaries of divisibility, so to speak, “deep” and does not allow control of all its interconnections and mediations) turns out to be compensated, consistent with the “perceptual-sensory” apparatus of the perceiving subject or device (which, basically the same). The mechanism of complexity self-compensation seems to us to be such a universal “universal” of the “subject-object” relationship that, perhaps, it should be attributed to the basic principles of system analysis.

The next question concerns the real possibilities of the control system to obtain the necessary data, process it within an acceptable time frame, develop a rational decision based on them and take effective measures for its implementation. All this should take place in real time and take into account the specifics of controlled objects and control subjects, their inertia, etc. Unfortunately, the listed operations for an object located in the exponential well, as a rule, are on or even beyond the capabilities of the control system. We can say that the complexity of the controlled object at a certain stage of evolution begins to exceed the capabilities of the control system. At the forefront here, as we see, are also the relations “subject-object”. The structure of these relations is quite bizarre, since what we call the “object of control” constantly “goes beyond” and begins to show the features of subjectivity, finding completely unexpected substantial resources for this. Thus, in order to conduct a correct system analysis of relations of this kind, we believe it is necessary to somewhat clarify the concept of a system, which consists in explicitly taking into account the action and interaction of the basic components of any system, namely: <Structure>, <Substance> and <Subject>. A detailed presentation of this clarification is presented in our works¹⁴.

So, in an arbitrary rather complex system, due to the limited nature of its resources, which always takes place, at a certain stage of evolution, a structure consisting of three zones begins to appear:

controlled (managed) zone;
uncontrolled (unmanaged) and
intermediate between them is weakly (conditionally) controlled (managed).

In terms of observability, it sounds like any sufficiently complex system has an observable (“light”) part, an unobservable (dark”), and an intermediate (“gray”) region

between them. Schematically represent this in the form of a partly “Manichaeian” picture:

LIGHT, GRAY AND DARK SIDES OF THE EVOLVING WORLD



The noted effect manifests itself in systems and subject areas that are completely different (practically, in all) in their origin and nature. Recently, K. V. Anokhin noted the effects of dark matter and dark energy even in the structure and operation of the human brain¹⁵. In our further work, we will show how these effects manifest themselves in the linguistic consciousness of a person and the human community. Now let’s move on to examples and start with the evolution of the physical world, which we already talked about in the first part.

According to the data of observations of the Planck space observatory published in March 2013, interpreted taking into account the standard cosmological model Lambda-CDM, the total mass-energy of the observable Universe consists of 4.9% ordinary (baryonic) matter, 26.8% from dark matter and 68.3% from dark energy. At the same time, ordinary matter from interstellar gas, masses of stars, planets, etc. The concept of dark matter was introduced to theoretically explain the problem of hidden mass in the effects of an anomalously high rotation speed of the outer regions of galaxies and gravitational lensing. The composition and nature of dark matter is currently unknown. The most likely candidates for the role of dark matter particles are hypothetical weakly interacting WIMP particles that do not participate in electromagnetic interactions. We mentioned dark energy above; it is responsible for the anomalous (“inflationary”) expansion of the Universe. According to some hypotheses, the expanding power of the dark energy will continue to increase indefinitely until it surpasses all other forces in the universe. In this scenario, dark energy eventually breaks all gravitationally bound structures of the Universe, then surpasses the forces of electrostatic and intranuclear interactions, breaks atoms, nuclei and nucleons and will

14 Широков В.А. Системні ефекти при лексикографічному описі мови. С.25 //System Analysis and Information Technologies 16-th International Conference SAIT 2014. Kyiv, Ukraine, May 26-30, 2014 Proceedings. Nataliya D. Pankratova (Ed.). В.А.Широков. Лингвистика и системный подход. Часть I. Бионика интеллекта. 2015. № 1 (84). С. 3–12. V. Shyrokov. “Language. Information. System” – К.; Наукова думка., 2021, 158 с.. Free access at: https://www.ulif.org.ua/system/files/blok-shirokov_book.pdf

15 Темная материя и темная энергия мозга.К В Анохин. (“Dark matter and dark energy of the brain”). <https://www.youtube.com/watch?v=HtZLZz1LLGM>

destroy the Universe in a Big Rip. Unless, of course, the World Mind chooses any other scenario for It.

Apparently, it is not given to us to become participants or even witnesses of such outstanding events. But science at the moment asserts precisely the described structure of the matter of the Universe, 95.1% consisting of dark matter and dark energy, which, in general, corresponds to the above evolutionary ideas, however, up to some terminological inversion, the essence of which becomes clear from the following table.

PHYSICAL TERM	EVOLUTIONARY ANALOGUE
Baryonic matter	Light matter
Dark matter	Gray (Deep) matter
Dark energy	Dark matter

Similar reasoning is applicable to the Internet – the World Wide Web (W3 – WWW), in which the following structure was built in the process of its evolution::

Usual Web ("Light Web") – is available to users using standard browsers;

Deep Web, Deep Net – a set of Web pages of the World Wide Web that are not indexed by search engines. Part of the deep web is the Deep Web, hidden web, which consists of web pages that are dynamically generated in response to queries against online databases:

DarkWeb, DarkNet – segments of the Network, although connected to the general Internet, but requiring special software tools for access. Darknet (DarkNet, also known as the "Hidden Network", "Dark Web", "Shadow Network", "Dark Web") is a hidden network, the connections of which are established only between trusted peers, sometimes referred to as "friends", using non-standard protocols and ports. This also includes the so-called Anonymous Network, which represents a system of unconnected virtual tunnels that provide encrypted data transfer. To designate the Darknet, the Anonymous symbol is sometimes used – the mask of Guy Fawkes, a participant in the so-called Gunpowder Plot (England, 1605).

In recent years, the term "Deep State" has gained considerable popularity, which is most often used to refer to a certain part of the state apparatus and structures affiliated with it, pursuing an independent, unofficial, law-bound policy in their group (clan, oligarchic, etc.) interests. The Deep State is often considered a conspiracy theory term, though it has taken on an almost official meaning lately. In the given meaning, the deep state acts as an opponent or even antagonist of the official state, although their interests and goals may well coincide. Often the border between them is almost indistinguishable. As for the Dark State, this phenomenon undoubtedly takes place, although the term has not yet gained popularity. In our understanding, the Dark State is illegal, including international entities, pursuing their policy with harsh and

even terrorist methods. As such, one can name the Islamic State (ISIS, ISIS, DAISH), the features of the Dark State have national mafia structures, etc.

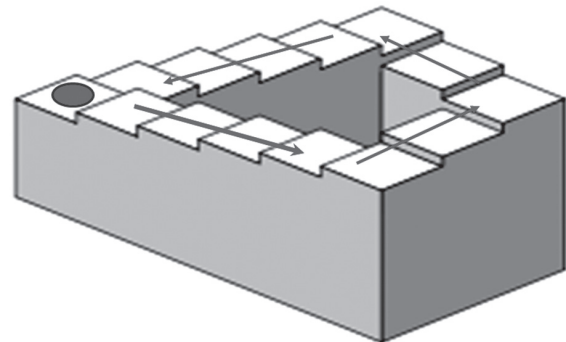
In the course of evolution in areas not regulated by law (such as crypto-currency activities, new types of weapons, cardinal private scientific and technological discoveries), one should also expect the emergence of structures capable of usurping important state functions and, de facto, playing the role of the Dark State.

It seems to us that a similar three-zone structure can be found in any object that is at a certain stage of evolution. It would be very interesting to analyze from our point of view the three-zone structure of the genetic mechanism, where, using the example of Amoeba dubia, we can expect the value of the σ -factor at the level of only one percent and, accordingly, the volume of gray and dark genetic matter at the level of 99%. We will not undertake this analysis here, leaving this work to competent professionals in the field of genetics, if our approach arouses their interest.

Let's move on to the analysis of the three-zone structure of money using the example of the dollar, the most advanced on the evolutionary ladder of the world currency.

2.7. "Dark matter" of dollar

If the image of a ladder came to us when we started talking about the evolution of the dollar, then let us continue this metaphor using the image of a ladder developed by Lionel and Roger Penrose¹⁶:



This paradoxical image – "Up the stairs leading down" – shows how to draw the impossible in the real world on a flat picture.

"Aber sieh, – as Franz Kafka noted in the famous parable "The trees", – sogar das ist nur scheinbar¹⁷"

The history of the Fed and further the IMF expressively says that it was all devoted to strengthening the influence of the dollar, expanding its geography, so that it became planetary, increasing its functionality, including purely speculative ones, which as a result far exceeded

16 GUIDE: The author himself put a red dot on the picture to indicate the beginning of the movement and red arrows to indicate its direction. So that the Reader does not go wrong by mistake.

17 «But see, even that is only appearance».

the functions of ordinary money. But then this story became more and more like running up the Penrose stairs. The more the possibilities of the dollar increased, the more universal the financial instruments associated with it became, the faster the “parasitic complexity” associated with it grew.

Its original monetary matter grew with new substances – numerous derivatives, special drawing rights and many others.

In the dollar, criminal relations began to be hidden more and more often and more sophisticatedly. The volume of a kind of “dark matter” and “dark energy” hidden in it has grown steadily and continues to grow.

The mentioned substances, without making a significant contribution to world development, give a large, constantly growing and in no way compensated burden on the world production and economic system.

Formally speaking, the dollar is currently in a superposition of three main states:

**ordinary legal money (light part);
derivatives, SDRs, cryptocurrencies, etc.(deep part);
illegal money or money with an illegal (criminal) origin (dark part).**

Symbolically, this can be represented using the formula:

$$\Psi = \alpha_{LI}\Psi_{LI} + \alpha_{DE}\Psi_{DE} + \alpha_{CR}\Psi_{CR}, \quad (6)$$

where Ψ_{LI} , Ψ_{DE} , Ψ_{CR} – functions of **light, deep and dark states**; α_{LI} , α_{DE} , α_{CR} – corresponding weight coefficients – numbers showing how much the corresponding state contributes to the superposition Ψ . The formula (6) demonstrates the logical structure of money, consisting of **Light, Gray (Deep) and Dark zones**. But unfortunately, there are no markers of “parasitic” factors in it, so finding the values of weight coefficients and establishing “golden proportions” for them, and even more so, effectively managing their value is almost impossible.

Truly, as they used to say in the old days: “Money does not smell!”.

For Rome, as everybody knows, this idyll ended by catastrophe.

On this major note, we conclude the presentation of the main content and proceed to concluding remarks.

CONCLUDING REMARKS

As can be seen from the above, the author’s evolutionary approach works on the entire range of the Universe. At least in all cases of its application, it gives more or less correct qualitative results.

And what is even more valuable, in our opinion, is that the main concept of the theory – complexity – is, in principle, a measurable quantity for which it is possible to develop even formalized quantitative models.

What else we consider valuable in this approach is that it allows, within the framework of a unified methodology, to analyze the phenomena and processes of both natural and social history. This gives reason to once again think about the fact that we live in a single, very connected, interdependent world. Therefore, as John Donne wrote: “...never send to know for whom the bell tolls; it tolls for thee”.

And finally.

The three-zone structure (*Light, Gray and Dark zones*) can be considered an established and universal fact of our evolutionary concept. And the “Corona” Crisis”, as a representative of a series of acute global crises of our time, really turns out to be connected with the *Dark Matter* of the financial-information-network mechanism. Unfortunately, this crisis is not the last one. No sooner had it ended than a new one began, in the epicenter of which was Ukraine.

The author would not like to draw eschatological pictures, but, as we have seen, Evolution is a consistent and unsentimental person. In the arsenal of its tools, catastrophes occupy a very honorable place.

Let’s hope that the World Mind will choose any other scenario for us.



УДК 004.75

DOI 10.30837/bi.2022.1(98).02

О.Є. Українець¹, С.Г. Удовенко², Л.Е. Чала¹, В.Л. Шергін¹¹ХНУРЕ, м. Харків, Україна, oleksandr.ukrainets@nure.ua,
ORCID iD: 0000-0001-8936-6591²ХНЕУ ім. С. Кузнеця, м. Харків, Україна, serhiy.udovenko@hneu.net,
ORCID iD: 0000-0001-5945-8647¹ХНУРЕ, м. Харків, Україна, larysa.chala@nure.ua, ORCID iD: 0000-0002-9890-4790¹ХНУРЕ, м. Харків, Україна, vadim.shergin@nure.ua, ORCID iD: 0000-0002-4388-8180

НЕЙРОМЕРЕЖЕВА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ З ВИКОРИСТАННЯМ КЛІТИННИХ АВТОМАТІВ

Пропонується технологія розпізнавання зображень, яка основана на нейромережеві обробці статичних та динамічних зображень з використанням клітинних автоматів (КА). Розглянуто принципи реалізації операцій обробки зображень з застосуванням КА (зокрема, операцій попередньої обробки, виділення контурів та розпізнавання). Проаналізовано особливості побудови згорткових нейронних мереж (ЗНМ) та їх використання для розпізнавання зображень. Розглянуто можливості інтеграції КА в нейромережеву модель класифікації та розпізнавання зображень. Наведено результати розробки та програмної реалізації нейромережевої технології розпізнавання зображень з використанням ЗНМ та КА. Здійснено моделювання запропонованої технології (на прикладі проблеми розпізнавання зорових об'єктів для користувачів з обмеженими можливостями зорового сприйняття) розробки та програмної реалізації нейромережевої технології розпізнавання зображень з використанням ЗНМ та КА.

ОБРОБКА ЗОБРАЖЕНЬ, КЛІТИННІ АВТОМАТИ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, APPLE VISION, РОЗПІЗНАВАННЯ ЗОРОВИХ ОБ'ЄКТІВ

Ukrainets O.Ye., Udovenko S.G., Chala L.E., Shergin V.L. Neural network technology of image recognition using cellular automata. An image recognition technology based on neural network processing of static and dynamic images using cellular automata (CA) is proposed. The principles of the implementation of image processing operations with the use of CA (in particular, pre-processing operations, contour selection and recognition) are considered. The peculiarities of the construction of convolutional neural networks (CNN) and their use for image recognition are analyzed. The possibilities of integrating CA into a neural network model of image classification and recognition are considered. The results of the development and software implementation of the neural network technology of image recognition using SNM and CA are given. Modeling of the proposed technology was carried out (on the example of the problem of recognizing visual objects for users with limited visual perception). development and software implementation of neural network technology for image recognition using SNM and CA.

IMAGE PROCESSING, CELLULAR AUTOMATA, CONVOLUTIONAL NEURAL NETWORKS, APPLE VISION, VISUAL OBJECT RECOGNITION

Вступ

В даний час отримали значний розвиток теоретичні та практичні дослідження проблеми розпізнавання образів у реальному часі [1, 2]. Розпізнавання образів вирішує два основних завдання: ідентифікацію та верифікацію. Завдання ідентифікації зазвичай полягає в послідовному порівнянні зображень виявленого об'єкту з зображеннями аналогічних об'єктів, що зберігаються в базі даних. Мета верифікації полягає в порівнянні отриманого результуючого зображення з відповідним еталоном (в разі його наявності) та кількісній оцінці їх близькості, що дозволяє віднести виявлений об'єкт до деякого класу.

При цьому виникає необхідність вирішення цілої низки складних завдань, зокрема, завдання динамічного пошуку та детекції об'єкта в часі та просторі з використанням наявних відеокамер. В разі використання камер сучасних мобільних телефонів вони мають бути оснащені додатковим спеціалізованим програмним забезпеченням (ПЗ) для оперативного

розпізнавання аналізованих зображень. Таке ПЗ може сприяти, наприклад, вирішенню проблеми розпізнавання зорових об'єктів для користувачів з обмеженими можливостями зорового сприйняття. При цьому мають застосовуватись інтелектуальні технології, що дозволяють користувачам компенсувати проблеми з зором шляхом часткового перенесення їх зорових функцій на обчислювальну техніку.

Основною перевагою такого підходу є доступність мобільних пристроїв у різних цінових сегментах. Смартфони з камерами, що дозволяють отримувати дані у вигляді зображень або відео є цілком доступними для більшості потенційних користувачів. При цьому слід відзначити варіативність використання різних за ціною мобільних пристроїв, що передбачають можливість завантаження та використання програмних засобів розпізнавання аналізованих зображень. Такі програмні засоби мають низку переваг та недоліків. До переваг можна віднести відносну простоту й швидкість їх розробки, оскільки для

мобільних пристроїв на основі сучасних операційних систем (зокрема, операційних систем iOS та Android) існують окремі мови програмування з великою кількістю бібліотек зразків зображень для навчання нейромережових моделей, що використовуються для розпізнавання та класифікації виявлених об'єктів.

Недоліками такого рішення є обмеженість якості зображень, що передаються через камеру мобільних пристроїв. Таке обмеження обумовлюється, зокрема, необхідністю постійного переміщення мобільного пристрою в процесі пошуку об'єкта, через що на якість отримуваної відеоінформації суттєво впливають динамічні викривлення зображень. У такому разі виникає необхідність застосування інтелектуальних засобів, які можуть нівелювати або зменшувати вплив викривлень та перешкод в процесі розпізнавання об'єктів. До таких засобів слід віднести використання клітинних автоматів (КА) в схемах нейромережового навчання систем розпізнавання.

У даній роботі пропонується технологія розпізнавання статичних та динамічних зображень, яка основана на використанні узагальненої моделі розпізнавання образів за допомогою поєднання можливостей КА та згорткових нейронних мереж (ЗНМ). Метою застосування цієї технології є підвищення точності та швидкості розпізнавання складних зображень.

Відповідно до поставленої мети, необхідно вирішити наступні завдання:

- аналіз принципів обробки зображень з використанням КА;
- аналіз особливостей використання ЗНМ для розпізнавання зображень;
- аналіз можливості інтеграції КА в нейромережову модель розпізнавання зображень;
- розробка та програмна реалізація нейромережової технології розпізнавання зображень з використанням ЗНМ та КА;
- моделювання запропонованої технології (на прикладі проблеми розпізнавання зорових об'єктів для користувачів з обмеженими можливостями зорового сприйняття).

1. Принципи обробки зображень з використанням клітинних автоматів

Останнім часом набувають розвиток дослідження, пов'язані з використанням клітинних автоматів для обробки зображень, оскільки можливість їх поєднання із нейромережовими технологіями надає можливість підвищити рівень надійності розпізнавання зорових об'єктів [3-5].

Клітинні автомати є дискретними динамічними системами, поведінка яких повністю визначається в термінах локальних залежностей. В значній мірі КА можуть бути застосовані і для опису великого класу безперервних динамічних систем, моделі яких,

наприклад, визначаються рівняннями в часткових похідних. У цьому сенсі клітинні автомати можуть функціонувати у просторі, що представлений рівномірною сіткою, кожна комірка якої містить декілька бітів даних. При цьому функціонування КА реалізується у дискретному часі за заданим набором правил (наприклад, таблицею, за якою будь-яка клітина на кожному кроці обчислює свій новий стан за станами її близьких сусідів). Клітинні автомати дозволяють побудувати корисні моделі для прикладних досліджень в природничих науках і комбінаторній математиці, які можуть бути застосовані для вивчення еволюції фізичних систем різного функціонального призначення. Слід зазначити, що клітинні автомати утворюють загальну парадигму паралельних обчислень. Клітинні автомати, важливе застосування в якості концептуальних і практичних моделей просторово розподілених динамічних систем, для яких фізичні системи є певними прототипами. Основний напрям дослідження клітинних автоматів – алгоритмічна вирішувальність поставлених задач (включно питання визначення початкових станів, при яких клітинний автомат матиме змогу їх вирішення).

Клітинні автомати можуть, зокрема, використовуватися для різних завдань обробки зображень, серед яких: геометричні перетворення, фільтрування шуму, виявлення особливостей, виявлення контурів. При цьому недостатньо вивченою можна вважати проблему сегментації та розпізнавання зображень різних типів за допомогою клітинних автоматів. Використання алгоритмів клітинних автоматів для сегментації зображень та їх подальшого розпізнавання має такі переваги:

- простота впровадження;
- паралельна реалізація;
- довільна кількість сформованих класів при проведенні сегментації (можлива сегментація зображень із двома мітками та з кількома мітками);
- варіативність правил переходу станів (зазвичай для цих правил використовуються зміни значень інтенсивності пікселів, але в механізм оновлення можуть бути включені і інші функції зображення, такі як текстура або краї);
- можливість роботи з зображеннями будь-якого розміру (на обчислювальну складність процесу безпосередньо не впливає розмір зображення або кількість функцій зображення) [6].

В обробці зображень зазвичай використовуються двовимірні КА. Пікселі зображення являють собою комірки КА і вони оновлюють свій стан на основі станів сусідніх комірок (пікселів). Кілька станів комірок КА дозволяють обробляти зображення сірого масштабу або кольорові зображення. Визначення правил, що застосовуються до комірок, щоб відповісти на певний запит при обробці зображень, є нетривіальним завданням.

Розглянемо зв'язок двовимірного КА з зображенням. Зображення I у загальному вигляді може бути описане як двовимірна функція:

$$I = f(x, y), \tag{1}$$

де x і y – просторові координати.

Амплітуда f в будь-якій парі координат (x, y) визначає інтенсивність I для сірих значень цифрового зображення. Цифрове зображення I представляється єдиним двовимірним цілим масивом для зображення сірого масштабу та серії з трьох двовимірних масивів для кожної кольорової смуги. Для зображення, що задається двовимірним масивом $m \times n$ пікселів, використовується двовимірна сіткова модель КА. При цьому зображення розглядається як двовимірний КА, де кожна комірка відповідає пікселю у зображенні, а інтенсивність пікселя представлена станом цієї комірки. Значення кольорів пікселів оновлюються синхронно в дискретному часі.

Топологію сітки двовимірного КА з околom фон Неймана наведено на рис. 1.

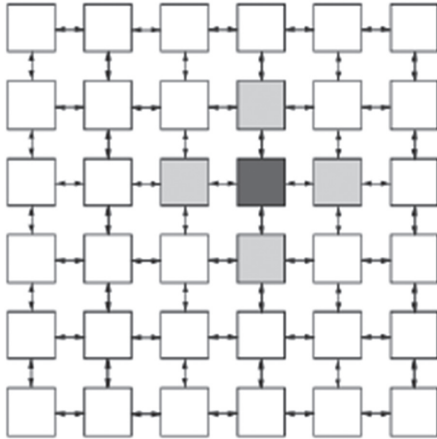


Рис. 1. Топологія сітки двовимірного КА з околom фон Неймана

Така топологія, де пов'язані лише найближчі сусіди аналізованої комірки (пікселя зображення), визначає окол фон Неймана з радіусом $r = 1$. Загалом, різні околи визначаються різними наборами зв'язків (рис. 2).

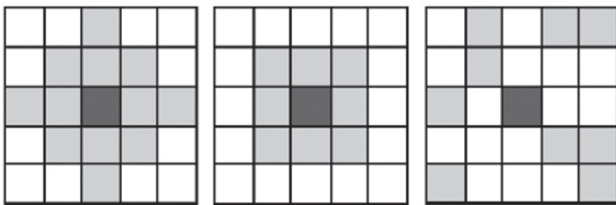


Рис. 2. Деякі типи околів двовимірного КА

На рис. 2 наведено (зліва направо) деякі типи найбільш вживаних околів двовимірного КА: фон Неймана (радіус 2), Мура (радіус 1) і випадковий.

Таким чином, двовимірний клітинний автомат складається з решітки квадратів $L = m \times n$, де кожен квадрат (клітина) представлений одним КА.

Змінні комірок у межах околів використовуються для оцінки нових значень оновленої комірки за допомогою функції (правила) переходу

Будь-яке правило переходу $\varphi(t)$ може бути записане у вигляді функції $\varphi(t)$, використовуючи стани $\sigma(t)$ всіх комірок в околах:

$$\sigma_{i,j}(t+1) = \varphi(\sigma_{i,j}(k, 1) \mid \sigma_{i,j}(k, 1) \in N), \tag{2}$$

де N – множина елементів околу (включно центральну комірку)

Правилом переходу $\varphi(t)$ може бути будь-яке поєднання арифметичних і логічних операцій, включаючи функції. Зазвичай правила переходу мають або форму таблиці переходу, що визначає залежність значення виходу від вхідних значень, або форму комп'ютерної програми.

Кількість правил стрімко зростає зі зростанням кількості сусідів та станів (табл. 1).

Таблиця 1

Залежність кількості правил двовимірного КА від кількості станів та сусідів

Кількість станів	Кількість сусідів	Кількість правил
2	2	16
2	3	256
2	5	4294967296

Навчання клітинних автоматів для завдання обробки зображень означає вибір оптимального набору правил серед усіх можливих правил, які найкраще відповідають конкретному завданню.

Отримати ефективні правила для досягнення бажаного результату дуже складно. Навіть для бінарних зображень кількість комбінацій всіх можливих правил є занадто велика. В цілому без перерахування всіх комбінацій оптимальний вибір правил не може бути гарантований. Необхідним є пошук поєднань між правилами з використанням оптимізаційних методів. В роботі [7], наприклад, запропоновано використовувати генетичний алгоритм (ГА) для вирішення проблеми виявлення контурів зображень (edge detection) за допомогою КА.

Ініціалізація ГА здійснюється шляхом побудови випадковим чином пакетів правил, отриманих з околів моделі КА на всьому зображенні. Для визначення елементів контуру ГА здійснює пошук серед поточних пакетів вхідного зображення правило, що відповідає околу аналізованого пікселя, та модифікує центральний піксель відповідно до визначеного переходу.

Правило переходу КА представляється за допомогою конкатенації клітин стану околу комірки до оновлення та додаванням її майбутнього стану після оновлення. Це правило (шаблон) трансформується як лінійний формат хромосоми.

Функція фітнесу, яка використовується для виявлення контурів, має такий вигляд:

$$F = 1 - (Err/L \times H), \quad (3)$$

де L та H – ширина та висота зображення; Err – кількість пікселів, де $ImageED \neq ImageIdealED$ (ED – Edge Detection).

Функція Err обчислює кількість неспівпадаючих точок на отриманому та ідеальному зображеннях. Далі генерується нова сукупність з застосуванням характеристик для ГА операторів селекції, кросовера та мутації. Цей процес триває до тих пір, поки не буде відбуватися подальше поліпшення функції фітнесу або до встановленої максимальної кількості ітерацій.

Таким чином, виявлення контурів зображень (або фрагментів зображень) можна інтерпретувати як еволюційний феномен множини КА, керований кращими пакетами правил КА, застосованих до зображення. Коли така система буде навчена на наборі зображень конкретного типу, вона зможе обробляти і інші зображення цього типу.

У випадку із зображеннями у відтінках сірого тренувальний процес для такої системи буде дещо ускладнений. В разі використання околів Мура, де комірки можуть мати 256 можливих інтенсивностей, для центрального пікселя зі значенням можливі 256^8 моделей сусідства. Якщо задане зображення сірого масштабу перетворюється на бінарне шляхом порогової декомпозиції, тоді результат попереднього розділу можна застосувати до отриманого зображення. Найбільш прийнятним методом такої декомпозиції слід вважати метод Оцу [8]. Завдяки властивості мінімізувати дисперсію в межах класу та максимізувати між дисперсією класу, метод Оцу здатний відображати максимальну кількість об'єктів, розташованих на зображенні, на порогове зображення.

В роботі [9] наведено результати навчання КА для виконання різних завдань обробки зображень. При цьому можливий набір правил зменшується з 256 до 51 на основі симетрії та рефлексії для центральних чорно-білих пікселів. Після цього проводиться навчання на основі скороченого набору правил.

Кожен зразок правила узгоджується із схемою сусідства пікселів зображення. Якщо він збігається, то колір центрального пікселя інвертується. Перед обробкою кожне правило перетворюється на 8-бітний рядковий шаблон і потім узгоджується з перетвореною схемою околу пікселя зображення. Для тренування системи мети використано метод послідовного плаваючого пошуку вперед (SFFS) з застосуванням цільової функції середньоквадратичної похибки між вхідним та цільовим зображеннями.

2. Особливості використання згорткових нейронних мереж для розпізнавання зображень

До найбільш розвинених та ефективних напрямків нейромережевої обробки статичних та динамічних зображень слід віднести застосування згорткових нейронних мереж (ЗНМ; англ. CNN – convolution neural networks). В загальному випадку мережі ЗНМ мають вхідний рівень (input), вихідний рівень (output) і кілька прихованих шарів.

Базову архітектуру ЗНМ, призначеної для класифікації та розпізнавання об'єктів (зокрема, зображень), наведено на рис. 3. В цій архітектурі присутні дві основні частини: виділення ознак об'єктів (feature extraction) та класифікація (classification). Виділення ознак є інструментом згортки, який розділяє та ідентифікує різні характеристики зображення для їх аналізу та подальшого використання на етапі класифікації.

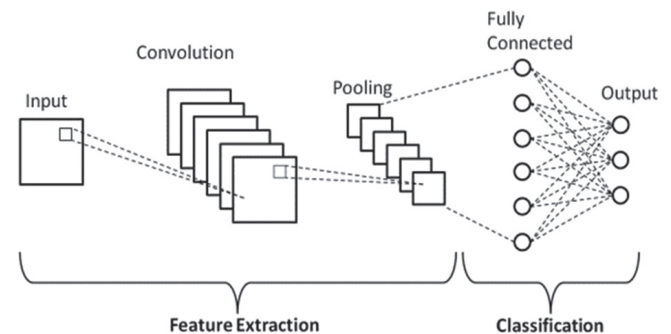


Рис. 3. Базова архітектура ЗНМ для класифікації зображень

Мережа виділення ознак ЗНМ складається з багатьох пар згорткових (convolution) та об'єднувальних (pooling) шарів. Ця мережа має на меті зменшити кількість ознак, присутніх у наборі даних. При цьому можуть створюватися нові функції, які узагальнюють існуючі функції, що містяться в оригінальному наборі функцій.

Згортковий шар є першим шаром, який використовується для виділення різних функцій із вхідних зображень. У цьому шарі виконується математична операція згортки між вхідним зображенням і фільтром (ядром), що є квадратною матрицею $n \times n$ (n – непарне число). Пересування ядра по вхідному зображенню дозволяє сформувати скалярний добуток між ядром і частинами вхідного зображення. Під час обчислення нового значення обраного пікселя ядро згортки «прикладається» своїм центром до даного пікселя. Навколишні пікселі так само накриваються ядром. Далі вираховується сума, де складовими є помножені значення пікселів на значення комірки ядра, що накрила даний піксель. Сума ділиться на коефіцієнт нормування (div) або на суму всіх елементів ядра згортки. Отримане значення є новим значенням обраного пікселя. Якщо застосувати згортку до кожного пікселя зображення, то буде отримано результат, що залежить від обраного ядра згортки.

Приклад роботи ядра згорткового шару наведено на рис. 4.

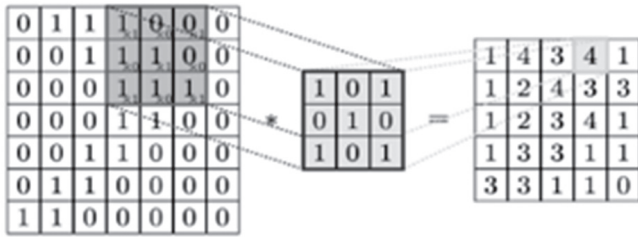


Рис. 4. Приклад роботи ядра згорткового шару

Результат операції згортки називається картою функцій, яка надає інформацію про зображення (наприклад, про кути та краї). Пізніше ця карта функцій передається на інші шари для вивчення особливостей вхідного зображення.

Ядро переміщується по всій області карти функцій і знаходить певні ознаки об'єктів. Наприклад, якщо мережу навчали на множині зображень осіб, то одне з ядер в процесі навчання може видавати найбільший сигнал в області очей, рота, інше ядро може виявляти інші ознаки. Якщо розмір ядра недостатній, то воно не зможе виділити потрібні ознаки, якщо ж занадто великий, то збільшується кількість зв'язків між нейронами. Зазвичай розмір ядра задають в діапазоні від 3×3 до 7×7. Ядро реалізує систему поділюваних ваг, що є однією з головних особливостей ЗНМ. У звичайній багат шаровій нейронній мережі існує занадто багато зв'язків (синапсів) між нейронами, що суттєво уповільнює процес детектування. У ЗНМ узагальнення ваг дозволяє зменшувати число зв'язків і знаходити одну і ту саму ознаку по всій області зображення.

Рівень згортки в ЗШМ передає результат на наступний рівень (шари об'єднання) після застосування операції згортки у вхідних даних. Згорткові шари в ЗШМ забезпечують безперервність просторового відношення між пікселями зображення.

Основна мета субдискретизуючих шарів об'єднання (Pooling Layers) – зменшити розмір згорткової карти функцій, щоб скоротити витрати на обчислення. Це виконується шляхом зменшення зв'язків між шарами та незалежно працює на кожній карті об'єктів. Ці шари зазвичай розміщуються одразу після згорткових шарів, тобто виходи згорткових шарів є входами для субдискретизуючих шарів мережі. Операції об'єднання скорочують розміри карт характеристик за рахунок використання деяких функцій для узагальнення субрегіонів, наприклад, взяття загального або максимального значення. Шари субдискретизації націлені на поступове зменшення розмірності даних і, отже, додаткове скорочення кількості параметрів, а також складність процедури моделі і, таким чином, контроль над питанням перенавчання. Існує ряд методів об'єднання (максимальне об'єднання,

середнє об'єднання, стохастичне об'єднання, спектральне об'єднання, об'єднання просторової піраміди, об'єднання по нормі L2 тощо).

Залежно від методу, який використовується, реалізуються кілька типів операцій об'єднання. Зокрема, операція Max Pooling полягає у визначенні найбільшого елемента з карти функцій. Ця операція розділяє вхідне зображення на набір прямокутників без перекриттів, і для кожної такої підобласті виводить її максимум. При цьому точне положення ознаки не так важливе, як її положення відносно інших ознак. Об'єднувальний шар сприяє поступовому скороченню просторового розміру представлення для зменшення кількості параметрів та об'єму обчислень у мережі, і відтак також для контролю перенавчання. Операція Average Pooling обчислює середнє значення елементів для зображення попередньо визначеного розміру. Загальна сума елементів у попередньо визначеному фрагменті визначається як об'єднання сум елементів. Рівень об'єднання зазвичай служить мостом між згортковим рівнем і рівнем класифікації. Ця модель ЗНМ узагальнює ознаки, витягнуті шаром згортки, і допомагає мережам розпізнавати функції незалежно. За допомогою цього також зменшуються обчислення в мережі. Найбільш поширеним є об'єднувальний шар шар з ядрами розміру 2×2, що застосовуються з кроком 2. В такому шарі знижується дискретизація кожного зрізу глибини входу в 2 рази як за шириною, так і за висотою, відкидаючи 75 % збуджень. В архітектурі ЗНМ інколи вставляють об'єднувальний шар між послідовними згортковими шарами.

Об'єднувальний шар, як і згортковий також має карти. Одним з важливих завдань цього шару є зменшення розмірності карт попереднього шару. Якщо на попередній операції згортки вже були виявлені деякі ознаки, то для операції об'єднання докладне зображення вже не потрібно, і воно суттєво ущільнюється. Слід відзначити, що фільтрація вже непотрібних деталей зменшує ризик виникнення перенавчання мережі. У процесі сканування ядром об'єднувального шару карти попереднього шару, скануюче ядро не перетинається на відміну від згорткового шару. Зазвичай, кожна карта має ядро розміром 2×2, що дозволяє зменшити попередні карти згорткового шару в 2 рази. Вся карта ознак поділяється на осередки 2×2 елементи, з яких вибираються максимальні за значенням.

Повнозв'язний рівень Fully Connected (FC) використовує вихідні дані процесу згортання та передбачає класифікацію зображення на основі ознак, виділених на попередніх етапах. Цей рівень складається з ваг і зміщень разом із нейронами та використовується для з'єднання нейронів між двома різними шарами. Ці шари зазвичай розміщуються перед вихідним

шаром і утворюють кілька останніх шарів архітектури ЗНМ.

У цьому випадку вхідне зображення з попередніх шарів вирівнюється та подається на повнозв'язний шар FC. Потім сплющений вектор проходить ще кілька шарів FC, де зазвичай виконуються операції математичних функцій. На цьому етапі починається процес класифікації. Доцільність з'єднання двох шарів полягає в тому, що два повністю з'єднані шари працюватимуть краще, ніж один з'єднаний шар.

Повнозв'язний шар зазвичай моделює складну нелінійну функцію, оптимізація якої сприяє підвищенню якості класифікації та розпізнавання об'єктів на вхідних зображеннях.

Нейрони кожної карти попереднього об'єднувального шару пов'язані з одним нейроном повнозв'язного шару. Таким чином, число нейронів повнозв'язного шару дорівнює числу карт об'єднувального шару, але при цьому тільки частина нейронів будь-якої з карт об'єднувального шару може бути пов'язана з першим нейроном прихованого шару. Частина, що залишилася, може бути пов'язана з другим нейроном прихованого шару.

Визначення кількості нейронів у шарі є важливим етапом побудови ЗНМ. На сьогодні не існує універсальних рекомендацій до вибору кількості прихованих шарів та кількості нейронів в них. Їх недостатня кількість не дозволяє мережі ефективно навчатися, а занадто велика кількість значно збільшує час навчання ЗНМ або призводить до її перенавчання (англ. *overfitting*). В разі виникнення перенавчання створена модель починає пояснювати лише приклади з навчальної вибірки, адаптуючись до них. Разом з тим вона не вчиться класифікувати приклади, які не залучалися до участі в навчанні, що зменшує здатність мережі до узагальнення можливих ситуацій класифікації зображень.

Найбільш поширеним емпіричним методом запобігання перенавчанню в ЗНМ є метод виключення (англ. *dropout*). Ідея метода полягає в тому, що в процесі навчання випадково із загальної мережі багаторазово видаляється деяка підмережа, оновлення ваг виконується тільки в межах цієї підмережі. Нейрони потрапляють в підмережу з ймовірністю p , яка має назву коефіцієнт дропаута. На наступному етапі на даних тренується лише редукована мережа, після чого видалені вузли повторно вставляються до ЗНМ з первинними вагами (рис. 5).

Таким чином, процедура виключення дозволяє уникати необхідності тренування всіх вузлів на всіх тренувальних даних, що знижує прояв ефекту перенавчання. Вона також значно покращує швидкість тренування та дозволяє здійснювати практичне поєднання моделей для ЗНМ.

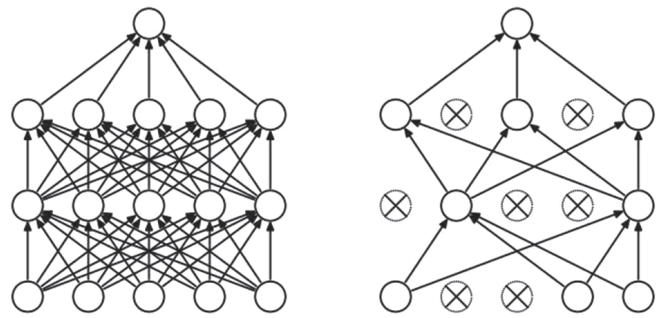


Рис. 5. Ілюстрація до виключення вузлів ЗНМ за методом Dropout

Однією з найважливіших задач побудови ЗНМ є вибір функцій активації. Вони використовуються для вивчення та апроксимації будь-якого виду безперервних і складних зв'язків між змінними мережі.

Зазвичай, в об'єднувальному шарі застосовується функція активації *ReLU* (англ. Rectified Linear Units – зрізані лінійні вузли). Цей шар застосовує ненасичувальну передавальну функцію $f(x) = \max(0, x)$. Він посилює нелінійні властивості функції ухвалення рішення і мережі в цілому, не зачіпаючи рецептивних полів згорткового шару. Для посилення нелінійності застосовуються й інші функції, наприклад, насичувальні гіперболічний тангенс $f(x) = \tanh(x)$, $f(x) = |\tanh(x)|$, сигмоїдна функція $f(x) = (1 + e^{-x})^{-1}$ та *Softmax* (нормалізована сігмоїда). Кожна з цих функцій має певні рекомендації до використання: зрізаному лінійному вузлові *ReLU* часто віддають перевагу перед іншими функціями, оскільки він тренує нейронну мережу доволі швидко без втрати точності узагальнення; сигмоїдна функція використовується, зазвичай, для бінарної класифікації, а *Softmax* – для багатокласової класифікації.

Крім стандартних архітектур ЗНМ останнім часом набули поширення нові способи побудови згорткових шарів, щоб підвищити ефективність навчання. Ці архітектури надають загальні архітектурні рекомендації для практиків машинного навчання, які необхідно адаптувати для вирішення різноманітних проблем комп'ютерного зору. Ці архітектури можна використовувати як екстрактори багатьох функцій для класифікації та розпізнавання зображень. До перспективних модифікованих архітектур ЗНМ слід, зокрема, віднести комбіновані архітектури з використанням можливостей КА.

3. Інтеграція клітинних автоматів в нейромереву модель розпізнавання зображень

Розглянемо ознаки певної еквівалентності клітинних автоматів та згорткових нейромеревих моделей.

Визначимо КА як динамічну систему з M можливими станами, яка оновлює своє значення на основі свого поточного значення та значення D інших клітин (безпосередніх сусідів у квадратній сітці).

Вочевидь існує M^D можливих вхідних рядків σ для функції КА, які ми індивідуально позначаємо як σ . Клітинний автомат реалізує оператор $G(\sigma)$, який повністю задає список правил переходу $\sigma \rightarrow m$, $m \in 0, 1, \dots, M - 1$. При цьому існує MM^D можливих унікальних $G(\sigma)$, кожен з яких реалізує свій набір правил. Наприклад, для гри в життя $M = 2$, $D = 9$, а $G(\sigma)$ – булева функція, яка відображає кожен з 512 можливих 9-бітних вхідних рядків в один біт. Визначальною особливістю КА є локальність динамічного оновлення правила, що гарантує обмеженість області правил, де розмір D обумовлює її верхню межу.

Визначимо конволюційну нейронну мережу як функцію, яка приймає на вхід багатоканальне зображення, до якого застосовується серія локальних згорток через «ядро», що навчається. Те саме ядро застосовується до всіх пікселів зображення, і кожен згортковий шар поєднує інформацію в межах фіксованого локального радіусу кожного пікселя вхідного зображення [8]. Багато стандартних конволюційних архітектур включають «об'єднуючі» шари, які знижують дискретизацію попереднього шару і тим самим консолідуєть локальну інформацію на все більших просторових масштабах; однак, всі ЗНМ, що розглядаються в цій статті, не включають кроки зниження дискретизації і, таким чином, зберігають повну розмірність вхідного зображення.

Розглянемо КА як рекурентні мережі mlpconv . Основна аналогія між клітинними автоматами і традиційними згортковими нейронними мережами виникає через локальність динаміки і одночасне темпоральне оновлення всіх просторових точок. Оскільки нейронні мережі в принципі можуть діяти як універсальні апроксиматори функцій, досить складна архітектура нейронної мережі може бути використана для повної апроксимації кожного правила $\sigma \rightarrow m$, що становить функцію КА $G(\sigma)$. Потім цей оператор для одного околу може бути реалізований як згортковий оператор у складі ЗНМ, що дозволяє синхронно застосовувати його до всіх околів пікселів на вхідному зображенні.

Таким чином, представлення КА за допомогою ЗНМ вимагає двох кроків: вилучення ознак для ідентифікації кожного з M^D вхідних випадків, що описують кожен окіл, а потім асоціацію кожного околу з відповідним вихідним пікселем. Можна показати, як представити будь-який КА за допомогою одного згортового шару, а потім і 1x1 згорткових шарів. Прийнятні ваги можна знайти аналітично, використовуючи аналіз самої КА, а не шляхом алгоритмічного навчання на вхідних даних. Таким чином, один із можливих варіантів визначає мережу, яка однозначно зіставляє кожен з M^D входів σ із шаблоном, у той час як інший варіант розглядає шари мережі як рівні деревоподібного пошуку, який ітеративно звужує

кожен з входів σ до бажаного виходу m .

Ключовим аспектом такого підходу є використання лише одного неединичного згортового шару, який є першим прихованим шаром в мережі. Поле цих конволюційних нейронів еквівалентно околу D КА. Усі наступні шари складаються з 1x1 згорток, які не консолідуєть жодної додаткової інформації про сусідів. Використання згорток 1x1 для реалізації логіки таблиці правил КА має підвищити ефективність мережі та зменшити обчислювальні витрати. Більше того, для такої реалізації КА мережа не потребує об'єднуючих шарів, що робить мережу еквівалентною установці невеликого конволюційного багатошарового перцептрон (mlpconv) на КА [8].

Розглянемо структуру та функції цього різновиду ЗНМ. Багатошаровий перцепційний згортковий шар (mlpconv) є комбінацією низькорівневих функцій за допомогою узагальненої лінійної моделі побудови ЗНМ. Втім реальні дані часто є нелінійними, тому доцільним є використання комбінації $\text{Conv} + \text{MLP}$ (згортка + багатошаровий перцептрон) замість операції лінійної згортки, щоб поліпшити здатність мережі отримувати абстрактні функції та здійснювати узагальнення. Mlpconv можна розглядати як мініатюрну багатошарову згорткову мережу, що міститься в локальному рецептивному полі кожної згортки. Каскадний процес інтеграції картки крос-функцій може дозволити мережі вивчати складні функції. Крос-канальний параметризований шар сприйняття еквівалентний згортці 1x1.

На рис. 6 та 7 наведено для порівняння загальні схеми побудови лінійного згортового шару ЗНМ та шару mlpconv .

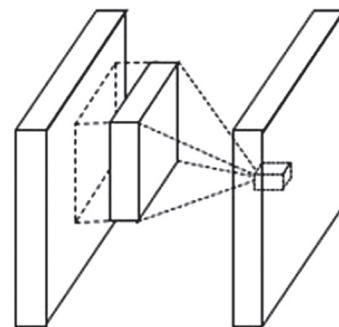


Рис. 6. Загальна структура лінійного згортового шару

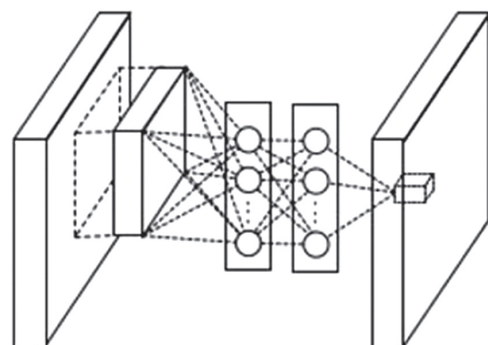


Рис. 7. Загальна структура шару mlpconv

Причина вибору MLP полягає в тому, що ця загальна функція апроксимації сумісна з базовою архітектурою ЗНМ, може бути поглиблена сама по собі (повторне використання функцій) і може бути навчена з використанням стандартного алгоритму зворотнього поширення помилки

Такий підхід до гібридного використання можливостей КА та конволюційного багат шарового перцептронну можна порівняти з використанням базових ЗНМ для розпаралелювання простих операцій двійкової арифметики, але він відрізняється від спроб використання менш поширених типів мереж з блоками сигма-пі, в яких кожен елемент у вхідному векторі розширюється попарною комбінацією помножених входів, а окремі вхідні біти можуть блокувати один одного [9].

4. Архітектура та принцип дії пропонованої системи розпізнавання зображень

Насамперед розглянемо алгоритмічну послідовність представлення довільного КА за допомогою ЗНМ.

Покажемо, як стандартний багат шаровий `mlpconv` перцептрон з активацією ReLU здатний представити довільний КА з M станами КА з кінцевою глибиною та обмеженою кількістю нейронів [8].

Розглянемо алгоритм зіставлення таблиці правил КА з мережею ЗНМ, що містить шар `mlpconv`.

Довільний двовимірний КА з M станами має спочатку бути перетворений у бінарне представлення. Для заданого $L \times L$ зображення сформуємо двійковий стек $L \times L \times M$ зображення для активації системи розпізнавання з використанням наступних дій, що реалізуються багат шаровим згортковим перцептроном `mlpconv`:

- згорнути вхідний шар за допомогою M різних згортки 1×1 фільтрів з одиничними вагами та зміщеннями, заданими як: $1, 0, -1, \dots, -(M-1)$; застосувати далі активацію ReLU;

- згорнути отримане зображення за допомогою згортки $M \ 1 \times 1$ фільтрів з нульовими зміщеннями. Кожен з перших $(M-1)$ згорткових фільтрів перевіряє послідовну пару $[1, -b, 0, \dots, 0], [0, 1, -b, 0, \dots, 0], [0, 0, 1, -b, 0, \dots, 0], \dots, [0, \dots, 0, 1, -b]$, де b – позитивна константа $b \ M/(M-1)$; останній згортковий фільтр $[0, \dots, 0, 1]$; застосувати знову активацію ReLU.

Таким чином, після виконання зазначених дій сформовано масив $L \times L \times (M-1)$, що відповідає одноразовому кодуванню значення кожного пікселя зображення в сітці $L \times L$. Далі шаблон зіставляє кожен із M^D можливих входів зі своїм відповідним вихідним значенням. Відзначимо, що кількість активних

станів в КА має бути задані заздалегідь і дорівнювати λM^D (λ – параметр Ленгтона) [17]. Далі оцінювання кількість шаблонів для відповідності (тобто кількість параметри) можна зменшити в λ раз. тому що тільки активні правила, які дають ненульовий результат, мають бути враховані.

Після формування масиву кодування значень пікселів необхідно:

- побудувати блок M^D , що містить $S \times S \times (M-1)$ згортки фільтрів, де S відповідає розміру околу двовимірного КА ($S = 3$ для стандартного КА з околом Мура). Кожен із фільтрів M^D просто відповідає зображенню можливого вхідного стану, що дорівнюють одиниці для кожного ненульового елементу зображення і великі від'ємні значення (більше $D(M-1)$) для кожного нульового елементу. Для випадків, коли $M > 2$, глибина кожного згорткового ядра дозволяє здійснити точне зіставлення різних ненульових значень.

- призначити зміщення кожному з M^D фільтрів на основі таблиці правил КА. Для входів $S \times S \times (M-1)$, які відповідають ненульовим значенням q , призначити зміщення $(q-1) - (L-1)$, де L – кількість ненульових вузлів в околі $L \ D(M-1)$. Це гарантує, що тільки точні збіги з правилом сформуєть додатні значення для елементів згортки мережі. Для вхідних даних, які мають відображатися на нуль, слід призначити будь-який зсув $L = D(M-1)$;

- застосувати функцію активації ReLU.

Наведений алгоритм обумовлює пропонований варіант архітектури `mlpconv` нейромережевої реалізації розпізнавання зображень, навченої на бінарних КА, яка може бути відображена такою схемою:

```
INPUT→[CONV3x3(1LAYER,100FILTERS)]→
[CONV1x1(11LAYER,100FILTERS)]→RELU→
[CONV1x1(LAYER,1FILTER)]→RELU→
FC(SOFTMAX)→OUTPUT.
```

Мережа складається з базової архітектури `mlpconv`, що відповідає одному згортковому шару 3×3 , за яким слідує змінна кількість згорткових шарів 1×1 [28]. Об'єднувальні шари при цьому не використовуються, а параметри шарів 3×3 і 1×1 навчаються разом. Останній шар здійснює зважене підсумовування, яке генерує передбачене значення наступного стану ділянки сітки КА. Включення останнього шару з класифікатором `softmax` прискорює навчання на бінарних КА, зменшуючи залежність збіжності від початкових ваг нейронів.

Мережа навчається за допомогою одного з оптимізаторів (зокрема, оптимізатора Адама з функцією втрат $L2$, з гіперпараметрами (швидкість навчання, початкові ваги і т.д.), оптимізованими за допомогою пошуку через сітку КА.

5. Розробка та тестування програмного додатку

Згідно зі схемою запропонованої нейромережевої технології розпізнавання зображень було здійснено її програмну реалізацію з використанням мови програмування Python і бібліотек TensorFlow та Keras, що дозволяють швидко побудувати і натренувати нейронну мережу. Програма передбачає можливість її інсталяції на мобільні пристрої.

Навчання програми оптимізатором Adam та її подальше тестування відбувалися з використанням навчальних вибірок, завантажених з датасету для розпізнавання продуктів Products-10K (dataset <https://products-10k.github.io/>).

Зокрема, одна з навчальних вибірок (Fruits) складалася з фотографій фруктів, розподілених за певними категоріями: яблуко (apple), банан (banana), апельсин (orange) тощо. Вона була поділена на множини train і valid (250 та 70 об'єктів відповідно).

Оскільки генерування нових навчальних даних не потребує великих обчислювальних витрат, для кожного етапу настроювання гіперпараметрів генерувався новий набір даних для валідації. Крім того, валідація проводилася з використанням випадково вибраних наборів правил КА, щоб гарантувати, що гіперпараметри мережі не були налаштовані на конкретні правила. Під час навчання того ж набору правил КА генерувався другий валідаційний набір даних, розмір якого становив 20% від розміру навчальних даних. Для контролю того, як вибір оптимізатора та функції втрат впливає на навчання, були створені та навчені реплікативні ансамблі мереж mlrconv, які мали ту саму форму мережі (12 шарів по 100 нейронів кожен), але іншу функцію втрат та оптимізатор, для яких були обрані різні оптимальні гіперпараметри. (таблиця 2).

Таблиця 2

Точність навченої мережі (mlrconv + CA) в залежності від типу оптимізатору

Оптимізатор	Точність (accuracy), %	Помилка (loss)
Adam	96,74	0.1521
SGD	82,65	0.2787
Rmsprop	94,35	0.1698

Дослідним шляхом було визначено оптимальну кількість епох навчання. Було проведено навчання мережі при різній кількості епох на тестовій вибірці (з оптимізатором Adam). Результати експерименту відображено у таблиці 3. Оптимальна кількість епох навчання склала 40.

Таблиця 3

Точність навченої мережі (mlrconv + CA) в залежності від кількості епох навчання

Кількість епох навчання	20	30	40	50
Точність (accuracy), %	91.55	92.68	96.74	95.12

Навчання успішно завершилося для всіх розглянутих наборів правил КА. Відзначимо, що використання архітектури згорткової мережі спрощує навчання повної таблиці правил. У результаті всі навчені мережі були працездатними, проте тривалість і динаміка навчання суттєво відрізнялися залежно від набору правил КА. Оскільки КА має тільки M^D можливих пар «вхід-вихід», а не необмежений простір входів, то навчання зупинялось, коли модель обробляла всі входи. Слід відзначити, що мережа mlrconv+КА може кодувати попереднє очікування для довільного входу, якщо вона була навчена на КА певного типу.

Зауважимо, що реальні мережі, навчені оптимізатором Adam, можуть знайти інші варіанти застосування mlrconv для реалізації довільних КА.

Для розробки мобільного додатку, що реалізує інсталяцію та застосування розглянутої технології розпізнавання зображень, було використано бібліотеку react-navigation.

Для отримання доступу до камери мобільного пристрою (в експериментах використовувався Apple iPhone 10) була використана бібліотека exocosm. Після отримання зображення з камери, перед тим, як відправити його на вхід до моделі нейромережі, передбачена обрізка зображення згідно з рамками, для чого використано бібліотеку expo-image-manipulator.

Під час практичного тестування розробленого мобільного додатку інколи виникали проблеми якості розпізнавання зображень, які обумовлені такими причинами:

- відсоток правильної класифікації знижується, коли освітлення є недостатнім, а на зображенні присутні тінюві плями;
- відсоток правильної класифікації знижується, коли в полі зору камери присутні однотипні об'єкти з близьких класів на недостатній відстані один від одного.

В умовах, коли мережа не була навчена на необхідних прикладах та не може точно класифікувати деякий об'єкт, то додаток його ідентифікує як найбільш подібний об'єкт, або оголошує про неможливість розпізнавання.

На рис. 8 наведено приклад розпізнавання зображення об'єкту з класу «Fruits».

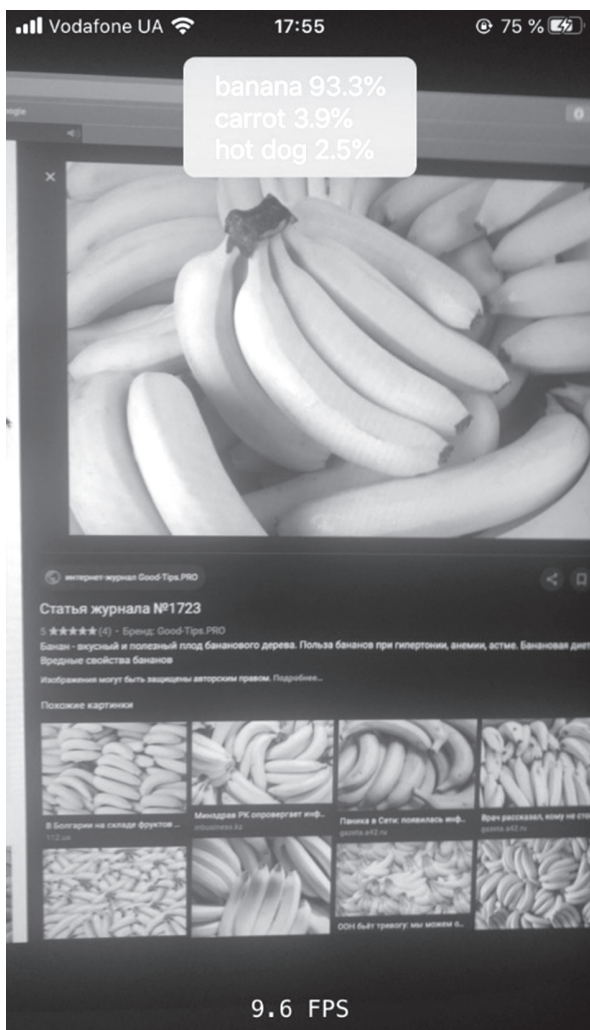


Рис. 8. Приклад розпізнавання зображення об'єкту з класу «Fruits».

Оскільки додаток орієнтовано на користувачів з обмеженими можливостями зорового сприйняття, то в ньому передбачено формування відповідного голосового повідомлення за результатами розпізнавання.

Висновки

До важливих завдань розпізнавання статичних та динамічних зображень слід, зокрема, віднести завдання динамічного пошуку та детекції об'єкта у часі та просторі з використанням наявних відеокамер

Перспективним напрямом вирішення цих завдань є застосування методів та засобів очислювального інтелекту. У статті наведено результати розробки та програмної реалізації нейромережевої технології розпізнавання зображень з використанням згорткових нейронних мереж та клітинних автоматів.

Побудову комбінованої нейронної мережі здійснено з частковим використанням бібліотеки Keras для мови програмування Python. Для формування наборів зображень в роботі використовувався датасету для розпізнавання продуктів Products-10K. Для розробки мобільного додатка, що реалізує встановлення

та застосування розглянутої технології розпізнавання зображень, була використана бібліотека react-navigation.

Результати проведеного тестування запропонованої системи розпізнавання зображень побутових об'єктів підтверджують її працездатність (зокрема, на тестовій вибірці середня точність прогнозу дорівнює 94.6%, що є цілком задовільним результатом для згорткових нейронних мереж з урахуванням малої кількості елементів вибірки). Розроблена система розпізнавання зображень з використанням мобільних пристроїв є гнучкою та розширюваною, що робить можливим її подальше удосконалення у напрямку підвищення точності розпізнавання та розширення функцій. Для підвищення точності системи доцільно збільшити навчальну вибірку, розширити діапазон представлених у ній зображень та додати у створену систему додаткові дані про особливості її застосування (наприклад, скорегувати набір гіперпараметрів мережі відповідно до побажань користувачів з обмеженим зоровим сприйняттям).

Список літератури:

- [1] *Schmidhuber, J.* (2015). "Deep Learning in Neural Networks: An Overview". *Neural Networks*. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. S2CID 11715509.
- [2] *Torlai, G., Mazzola, G., Carrasquilla, J. et al.* Neural-network quantum state tomography. *Nature Phys* 14, 447–450 (2018). <https://doi.org/10.1038/s41567-018-0048-5>
- [3] *Жежерун О.П., Калітовський Б.В.* (2019). Використання клітинних автоматів для вирішення задач фільтрації шумів та виявлення контурів зображень. *Наукові записки НаУКМА. Комп'ютерні науки*. Том 2. 66 – 72.
- [4] *Shukla, A. P.* (2016). Training Cellular Automata for Image Edge Detection. *Romanian Journal of Information Science and Technology*, 19, 4, 338–359.
- [5] A review of cellular automata models for crowd evacuation / Li Yang et al. *Physica A*: (2019). *Statistical Mechanics and its Applications*. Vol. 256. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0378437119303528> (accessed 05.05.2021).
- [6] *Diosan, Laura & Andreica, Anca & Enescu, Alina.* (2017). The Use of Simple Cellular Automata in Image Processing. *Studia Universitatis Babeş-Bolyai Informatica*. 62. 5-14. 10.24193/subbi.2017.1.01.
- [7] *Rosin, Paul & Sun, Xianfang.* (2011). Cellular Automata as a Tool for Image Processing. 10.1142/9789814343008_0012.
- [8] *William Gilpin.* Cellular automata as convolutional neural networks. *Physical Review E*, 100(3), Sep 2019. ISSN 2470-0053. doi: 10.1103/physreve.100.032402. URL <http://dx.doi.org/10.1103/PhysRevE.100.032402>.
- [9] *Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin* (2020). Growing neural cellular automata. doi: 10.23915/distill.00023. URL <https://distill.pub/2020/growing-ca>.

Надійшла до редколегії 16.02.2022



М.М. Корабльов¹, О.О. Фомічов², Д.В. Васюк³

¹доктор технічних наук, професор кафедри комп'ютерних інтелектуальних технологій та систем, Харківський національний університет радіоелектроніки, mykola.korablyov@nure.ua, ORCID ID: 0000-0002-8931-4350

²кандидат технічних наук, старший викладач кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, oleksandr.fomichov@nure.ua, ORCID ID: 0000-0001-9273-9862

³студент групи КІТм-20-1, Харківський національний університет радіоелектроніки, dmytro.vasiuk@nure.ua, ORCID ID: 0000-0003-4730-3378

ПОШУК ПРАЦЕЗДАТНИХ PROXY-АДРЕСІВ З ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖЕВОВОГО ПІДХОДУ

Для підключення користувача до Інтернету можна використовувати проху-сервери, які забезпечують різні рівні функціональності, безпеки та конфіденційності, і які можна налаштувати як веб-фільтри або брандмауери, що захищають комп'ютер від інтернет-загроз. Під працездатністю проху-адресів мається на увазі можливість відправляти запити на різні ресурси або сервіси та отримувати відповіді. Якщо запит був відправлений, а відповідь так і не отримана, можна зробити висновок, що проху-адреса не працездатна. У випадку отримання відповіді можна зробити висновок, що проху-адреса працездатна. Розглянуті фактори, які можуть впливати на відправлення запиту та отримання відповіді від ресурсу. Аналіз розглянутих факторів показав, що неможливо однозначно зробити висновок про працездатність проху-адреси, якщо брати до уваги лише факт наявності або відсутності відповіді від ресурсу. Виділені умови для підтвердження або спростування працездатності проху-адреси, які взяті за вхідні дані моделі, яка пропонується для оцінки працездатності проху-адреси. В якості моделі використана нейронна мережа у вигляді тришарового перцептрон, навчання якої виконується методом зворотного розповсюдження помилки. Створену модель було перевірено на визначення стану працездатності проху-адресів за допомогою тестових наборів даних. На основі результатів досліджень та проведених експериментів був створений додаток, який виконує пошук в Інтернеті ресурсів з проху-адресами та перевіряє їх на працездатність.

PROXY-СЕРВЕР, PROXY-АДРЕСА, ЗАПИТ, ВІДПОВІДЬ, ПРАЦЕЗДАТНІСТЬ, КОРИСТУВАЧ, РЕСУРС, НЕЙРОННА МЕРЕЖА, ДОДАТОК

Korablyov M.M., Fomichev O.O., Vasyuk D.V. Finding operability proxy addresses using a neural network approach. Proxy servers can be used to connect the user to the Internet, which provides different levels of functionality, security, and privacy, and can be configured as web filters or firewalls that protect the computer from Internet threats Under the health of proxy addresses, we mean the ability to send requests to various resources or services and receive responses. If the request was sent and no response was received, it can be concluded that the proxy address is not working. If a response is received, it can be concluded that the proxy address is operational. Factors that can influence, sending a request and receiving a response from a resource are considered. An analysis of the considered factors showed that it is impossible to unambiguously conclude the performance of a proxy address if we take into account only the fact of the presence or absence of a response from the resource. The conditions for confirming or refuting the performance of a proxy address are identified, which are taken as input data of the model proposed for assessing the performance of a proxy address. As a model, a neural network in the form of a three-layer perceptron was used, the training of which is performed by the method of error backpropagation. The created model was tested to determine the health status of proxy addresses using test datasets. Based on the results of research and experiments, an application was created that searches the Internet for resources with proxy addresses and checks their performance.

PROXY-SERVER, PROXY-ADDRESS, REQUEST, ANSWER, OPERABILITY, USER, RESOURCE, NEURAL NETWORK, APPENDIX

Вступ

Сьогодні неможливо представити сучасний світ без використання Інтернету, який надає користувачеві різні послуги. Підключення користувача до Інтернету може виконуватися безпосередньо через Інтернет-провайдер, або використовувати додатково транзитний веб-сервер – проху-сервер (посередник), який залежно від потреб використання забезпечує різні рівні функціональності, безпеки та конфіденційності [1]. Роль проху-сервера у схемі передачі даних між користувачем та кінцевим ресурсом наведе на рис. 1.



Рис. 1. Роль проху у схемі передачі даних між користувачем та кінцевим ресурсом

Проху-сервер є системою або маршрутизатором, який забезпечує шлюз між користувачами та Інтернетом, що допомагає запобігти проникненню кібер-зловмисників у приватну мережу. Його можна налаштувати як веб-фільтр або брандмауер, що захищає комп'ютер від інтернет-загроз, наприклад, таких як шкідливе програмне забезпечення. Проху-сервер може знаходитися на локальному комп'ютері користувача або в будь-якій точці між комп'ютером користувача та цільовими серверами в Інтернеті.

Існують апаратні та програмні версії проху-серверів [2]. Апаратні з'єднання знаходяться між комп'ютерною мережею та Інтернетом, де вони отримують, надсилають та пересилають дані з Інтернету. Програмні проху-сервери зазвичай розміщуються у постачальника або знаходяться у хмарі. Тоді необхідно завантажити та встановити відповідну програму на комп'ютер, яка полегшує взаємодію з проху-сервером.

Коли комп'ютер підключається до Інтернету, він використовує IP-адресу, яка схожа на поштову адресу, повідомляючи вхідні дані, куди йти, і позначаючи вихідні дані зворотною адресою для автентифікації інших пристроїв. Проху-сервер – це, по суті, комп'ютер в Інтернеті, який має власну IP-адресу. Оскільки проху-сервер має власну IP-адресу, він діє як посередник між комп'ютером та Інтернетом. Комп'ютер знає цю адресу, і коли подається веб-запит до Інтернету на використання ресурсу, він надсилається на проху-сервер, який оцінює запит та виконує необхідні мережеві транзакції: отримує відповідь від веб-сервера і пересилає дані зі сторінки до браузера комп'ютера. Це дозволяє спростити, контролювати складність запиту або надати додаткові переваги, такі як балансування навантаження, конфіденційність або безпека.

1. Вибір типу проху-сервера

Перш ніж почати пошук проху-серверів, необхідно зрозуміти, що існує кілька доступних їх типів, кожен з яких виконує різні функції [3-5]. Тому важливо вибрати правильний проху-сервер для ваших потреб. Слід зазначити, що не всі проху-сервери працюють однаково. Важливо зрозуміти, які функції ви отримуєте від проху-сервера, і переконатися, що даний сервер відповідає вашим цілям використання. Найчастіше проху-сервери застосовуються для таких цілей [2]:

- забезпечення доступу комп'ютерів локальної мережі до Інтернету;
- кешування даних: для зниження навантаження на канал у зовнішню мережу та прискорення отримання клієнтом запитаної інформації, якщо часто відбуваються звернення до тих самих зовнішніх ресурсів;
- стиснення даних: проху-сервер завантажує інформацію з Інтернету та передає інформацію

кінцевому користувачеві в стислому вигляді для економії зовнішнього мережевого трафіку клієнта;

- захист локальної мережі від зовнішнього доступу: наприклад, можна налаштувати проху-сервер так, що локальні комп'ютери будуть звертатися до зовнішніх ресурсів тільки через нього, а зовнішні комп'ютери не зможуть звертатися до локальних взагалі (вони бачать тільки проху-сервер);

- обмеження доступу з локальної мережі до зовнішньої: наприклад, можна заборонити доступ до певних веб-сайтів, обмежити використання Інтернету локальним користувачам, встановити квоти на трафік чи смугу пропускання, фільтрувати рекламу та віруси;

- анонімізація доступу до різних ресурсів: проху-сервер може приховувати інформацію про джерело запиту або користувача. У такому разі цільовий сервер бачить лише інформацію про проху-сервер, наприклад його IP-адресу, але не має можливості визначити справжнє джерело запиту; існують також спотворюючі проху-сервери, які передають цільовому серверу неправдиву інформацію про справжнього користувача;

- обхід обмежень доступу: використовується, наприклад, користувачами країн, де доступ до деяких ресурсів обмежений законодавчо та фільтрується.

Використання проху-серверів має ряд переваг [6]:

- посилений захист: проху-сервер може діяти як брандмауер між вашими системами та Інтернетом;
- приватний перегляд, прослуховування та покупки;
- можна використовувати для обмеження доступу до певних сайтів.

Разом з тим, у проху-серверів є недоліки [6]:

- відстеження: дані кешу, які використовують проху, можуть запам'ятовувати всю особисту інформацію, включаючи паролі, і є ймовірність, що співробітники, які працюють під проху, зловживають цією інформацією;
- безпека: хоча проху забезпечують переваги анонімності, їм не вистачає шифрування;
- вартість: налаштування та обслуговування проху-сервера можуть бути дорогими;
- конфігурації проху попередньо запрограмовані для однієї конкретної мети.

При підключенні комп'ютера до Інтернету через проху-сервер важливим являється визначення його працездатності, що впливає на якість послуг, які надаються. Враховуючи наведені властивості проху-серверів необхідно провести аналіз підходів, які використовуються для аналізу їх працездатності, факторів, які на це впливають, а також умов для визначення працездатності проху-адресів.

2. Працездатність проху-сервера

Під працездатністю проху-серверу мається на увазі можливість відправляти запити на різні ресурси або сервіси та отримувати відповіді (request/response) [7]. Якщо запит був відправлений, а відповідь так і не отримана, можна зробити висновок, що проху-сервер, який використовувався в запиті, не працездатний. У випадку отримання відповіді можна зробити висновок, що проху-сервер працездатний.

Але все не так однозначно. З логічної точки зору, якщо результат отриманий, то проху-сервер працездатний, якщо результат був не отриманий, то ні (1 або 0). Але не у всіх випадках. На відправлення запиту, з використанням проху-адреси, та отримання відповіді від ресурсу можуть впливати наступні фактори [8-10]:

1) connection timeout – проміжок часу, протягом якого слід встановити зв'язок між відправником запиту та ресурсом відповідача;

2) request timeout – час очікування відповіді від ресурсу. Після встановлення з'єднання між відправником та ресурсом, відправник повинен періодично інформувати ресурс про те, що він ще там, посилавши інформацію на цей інформаційний ресурс. Якщо відправник не може надсилати будь-яку інформацію на ресурс у вказаний час, ресурс просто знімає це підключення, оскільки він «думає», що відправника більше немає, щоб не витратити марно ресурси;

3) завантаженість проху-адреси – кількість користувачів, що одночасно використовують одну й ту ж проху-адресу. Від цього фактору залежать двоє попередніх факторів;

4) завантаженість ресурсу – максимальна кількість одночасно оброблюваних запитів;

5) швидкість проху-адресу – залежить від швидкості Інтернету;

6) нестабільне з'єднання з інтернет мережею.

Якщо брати до уваги той факт, що використовуються проху-адреси, які знаходяться у вільному доступі, можна доповнити список факторів наступними [8-10]:

1) життєвий період – зазвичай проху-адреси з вільного доступу мають життєвий період в 2-3 дні. Після третього дня ймовірність працездатності проху-адреси майже нульова;

2) можлива наявність авторизації на проху-сервері;

3) підробка проху-адреси – деякі сервіси дають безкоштовно користуватись їх проху-адресами, але заздалегідь разом з працездатними адресами йдуть і підробки;

4) старі проху-адреси – ресурс, з якого беруться адреси, може рідко оновлювати свій список.

Враховуючи ці фактори, можна відзначити, що неможливо однозначно зробити висновок про працездатність проху-адреси, якщо брати до уваги

всього лише факт наявності відповіді від ресурсу або її відсутності. Тому для пошуку працездатних проху-адресів необхідно застосовувати формалізований підхід, заснований на використанні математичної моделі, яка б дозволяла більш об'єктивно і точно оцінювати працездатність проху-адресів.

Розглянемо умови для визначення працездатності проху-адресів. На основі факторів, які можуть впливати на відсутність відповіді, можна виділити наступні умови для підтвердження або спростування працездатності проху-адреси [6-8]:

1) наявність відповіді від ресурсу;

2) чи був проху-адрес доданий раніше, чи поточний адрес був доданий вперше;

3) наявність відповіді в попередній раз;

4) перевірка дати додання в базу даних, дата додання може більше або менше трьох днів;

5) частота оновлювання проху-адреси, як часто проху-адреса потрапляє в базу даних;

6) відсоткове відношення між кількістю успішно отриманих відповідей від ресурсу та кількістю невдалих, під час перевірки проху-адреси;

7) відсоткове відношення між кількістю успішно отриманих відповідей від ресурсу та кількістю невдалих, під час відправки запитів від клієнта;

8) перевірка на кількість невдало отриманих відповідей поспіль.

Визначені умови будемо вважати за вхідні дані моделі, яка буде використовуватися для оцінки працездатності проху-адреси.

3. Модель оцінки працездатності проху-адреси

Для оцінки працездатності проху-адреси можна використати різні моделі, що основані на застосуванні тих чи інших методів інтелектуальної обробки інформації, основними з яких є такі [11]:

1) штучні нейронні мережі (НМ), перевагою яких є здатність представляти будь-яку обмежену безперервну функцію з будь-якою невеликою помилкою апроксимації;

2) нечітка логіка, яка дозволяє відображати вхідні дані або змінні в проблемі рішення так, як люди міркують про них;

3) експертні системи, які застосовуються для опису проблеми і використовують інтелект одного або кількох ідентифікованих експертів;

4) еволюційні обчислення, які характеризуються здатністю адаптації з метою пристосування до навколишнього середовища, моделюючи появу, виживання та вдосконалення популяції індивідів;

5) мультиагентні системи, що складаються з груп агентів з різними цілями та завданнями, які мають певні характеристики, і являються активною сферою досліджень у складних додатках.

Для створення моделі оцінки працездатності гроху-адресів будемо використовувати неймережвий підхід, який є ефективним для розв'язання широкого спектра задач інтелектуального аналізу даних, до яких можна віднести і задачу аналізу гроху-адресів. Існує велика кількість типів НМ, кожна з яких використовується для розв'язання відповідної задачі [12]. В якості моделі для оцінки працездатності гроху-адреси достатньо використати просту НМ у вигляді тришарового перцептрона (рис. 2).

Вхідні дані

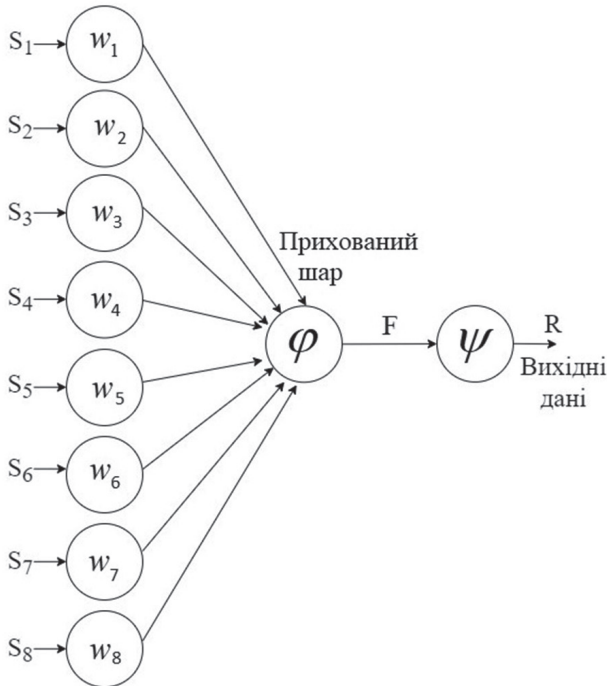


Рис. 2. Неймережева модель для оцінки працездатності гроху-адресів

Вхідний шар складається з вхідних нейронів, на які подаються значення ($S_1 - S_8$) конвертованих умов працездатності гроху-адреси, кожне значення яких дорівнює 0 або 1. З вхідних нейронів інформація з вагами (синапсами) передається в прихований шар, який складається з одного нейрона і реалізує нелінійну функцію:

$$F = \phi\left(\sum_{j=1}^n w_j \cdot s_j\right)$$

з сигмоїдальною функцією активації

$$\phi(u) = \frac{1}{1 + e^{-\lambda \cdot u}},$$

де $S = \{s_1, s_2, \dots, s_n\}$ – вхідні змінні; $W = \{w_1, w_2, \dots, w_n\}$ – набір ваг, що утворюють пам'ять нейрона.

Прихований шар, в свою чергу, передає інформацію у вихідний шар, який складається з одного нейрона, в якості функції активації ψ якого обрана бінарна порогова функція:

$$R = \psi(\phi(w, s)) = \begin{cases} 1, & \text{якщо } \phi(w, s) > 0,5; \\ 0, & \text{якщо } \phi(w, s) \leq 0,5. \end{cases}$$

На виході перцептрона отримуємо значення працездатності гроху-адреси: 1 – гроху-адреса працездатна, 0 – ні.

Розглянемо вхідні змінні НМ, яких вісім, і умови їх конвертації у відповідні значення.

S_1 – наявність відповіді від ресурсу. Якщо під час тестування гроху-адреси відповідь з ресурсу була отримана, значенню буде присвоєно 1, якщо ні – 0.

S_2 – чи є дана гроху-адреса новою, чи ні. Якщо гроху-адреса нова, то значенню буде присвоєно 1, якщо ні – 0.

S_3 – наявність відповіді під час перевірки в попередній раз. Якщо відповідь була отримана в попередній раз, значенню буде присвоєно 1, в іншому випадку – 0.

S_4 – чи пройшло з початку додання гроху-адреси до бази даних три дні, чи ні. Якщо проміжок часу між датою додання та поточною більше або дорівнює трем дням, значенню буде присвоєно 1, якщо ні – 0.

S_5 – наявність гроху-адреси на різних джерелах. Якщо кількість джерел у сумі складає 35 і більше відсотків, значенню буде присвоєно 1, якщо ні – 0.

S_6 – кількість разів, коли було отримано відповідь від ресурсу під час тестової перевірки гроху-адреси. Якщо кількість разів у сумі складає 30% і більше, значенню буде присвоєно 1, якщо ні – 0.

S_7 – кількість разів, коли було отримано відповідь від ресурсу під час відправки запитів клієнтом. Якщо кількість разів у сумі складає 7 і більше відсотків, значенню буде присвоєно 1, в іншому випадку – 0.

S_8 – наявність п'яти поспіль не отриманих відповідей від ресурсу. Якщо умова буде виконана, значенню буде присвоєно 1, якщо ні – 0.

Вихідною змінною НМ є значення працездатності гроху-адреси R : 1 – працездатна, 0 – непрацездатна.

4. Експериментальні дослідження

Щоб перевірити правильність функціонування НМ, було задано набір з 10 прикладів тестових вхідних значень, які наведені в табл. 1, де ($N_1 - N_{10}$) – номери тестових даних, ($S_1 - S_8$) – вхідні дані, R – результат роботи НМ.

Таблиця 1

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_2	0	0	0	0	0	1	1	0	1
N_3	0	0	0	1	0	1	1	0	0
N_4	0	0	0	1	1	1	1	0	1
N_5	0	0	1	1	0	0	0	0	1
N_6	0	1	0	1	0	0	0	0	0
N_7	1	0	0	0	0	0	0	0	1

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	R
N_8	0	0	0	0	0	1	1	1	0
N_9	0	0	1	0	1	1	1	1	0
N_{10}	1	0	0	1	0	0	0	0	1

Для тестування НМ на вхід подавався набір тестових даних ($N_1 - N_{10}$), які вказані в табл. 1. Ваги для кожного входу були обрані випадковими:

$$w_1 = 0.6294, w_2 = 0.8116, w_3 = -0.7460, \\ w_4 = 0.8268, w_5 = 0.2647, w_6 = -0.8049, \\ w_7 = -0.4430, w_8 = 0.0938.$$

Після задання ваг виконувалася симуляція НМ, результати якої представлені в табл. 2, де Σ_w – сума ваг кожного вхідного значення, F – результат на виході нейрона прихованого шару, R – вихідне значення НМ.

Таблиця 2

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Σ_w	F	R
N_1	0	0	0	0	1	1	0	0	3.5146	0.8431	1
N_2	0	0	0	0	0	1	1	0	4.6324	0.9904	1
N_3	0	0	0	1	0	1	1	0	0.0561	0.5140	1
N_4	0	0	0	1	1	1	1	0	-4.4086	0.0120	0
N_5	0	0	1	1	0	0	0	0	4.7822	0.9917	1
N_6	0	1	0	1	0	0	0	0	0.5608	0.6366	1
N_7	1	0	0	0	0	0	0	0	5.1370	0.9942	1
N_8	0	0	0	0	0	1	1	1	-0.5044	0.3765	0
N_9	0	0	1	0	1	1	1	1	4.3895	0.9877	1
N_{10}	1	0	0	1	0	0	0	0	0.5607	0.6366	1

З проведеної симуляції (табл. 2) можна зробити висновки, що її результати не збігаються тестовими вхідними даними (табл. 1). Для отримання потрібних результатів необхідно навчити НМ. З цією метою був складений набір з 30 комбінацій навчальних значень, за якими проводилося навчання НМ методом зворотного розповсюдження похибки. Були отримані нові, більш коректні значення ваг:

$$w_1 = 0.6294, w_2 = 0.8116, w_3 = -0.7460, \\ w_4 = -4.8160, w_5 = -4.8074, w_6 = -4.8077, \\ w_7 = -4.8076, w_8 = -4.8187,$$

та проведена симуляція на тестових даних (табл. 1). Результати роботи НМ представлені в табл. 3.

Таблиця 3

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Σ_w	F	R
N_1	0	0	0	0	1	1	0	0	-3.5812	0.0016	0
N_2	0	0	0	0	0	1	1	0	-6.4568	0.0016	0
N_3	0	0	0	1	0	1	1	0	-11.2547	1.2946e-05	0
N_4	0	0	0	1	1	1	1	0	-14.4831	5.1292e-07	0
N_5	0	0	1	1	0	0	0	0	8.3244	0.9998	1
N_6	0	1	0	1	0	0	0	0	10.5064	1.0000	1
N_7	1	0	0	0	0	0	0	0	15.1222	1.0000	1
N_8	0	0	0	0	0	1	1	1	-23.1957	8.4379e-11	0
N_9	0	0	1	0	1	1	1	1	-13.3018	1.6716e-06	0
N_{10}	1	0	0	1	0	0	0	0	10.3242	1.0000	1

Із результатів симуляції (табл. 3) можна зробити висновки, що НМ ще не обчислює потрібні значення і потребує більшої кількості навчальних наборів. Додатково було використано ще 60 комбінацій навчальних значень, за якими проводилося до навчання НМ і були отримані нові значення ваг:

$$w_1 = 15.1222, w_2 = 15.3044, w_3 = 13.1223, \\ w_4 = -4.7979, w_5 = -3.2284, w_6 = -3.2284, \\ w_7 = -3.2284, w_8 = -16.7389.$$

Результати роботи НМ після проведеної симуляції представлені в табл. 4. Видно, що отримані результати (табл. 4), збігаються з тестовими (табл. 1). Таким чином, можна зробити висновок, що НМ успішно завершила навчання.

На основі проведених досліджень по навчанню і тестуванню НМ, що оцінює працездатність проху-адресів, побудовано сервіс «ProхuClassifier», який приймає масив класів і конверторів умов. Дані класи конвертують поточний стан проху-адреси в зрозумілі для нього значення – 0 або 1, після чого відбувається процес аналізу проху-адреси на її працездатність. У разі успішного підтвердження буде повернуто 1, в іншому випадку – 0.

Таблиця 4

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	Σ_w	F	R
N_1	0	0	0	0	1	1	0	0	-1.1027	0.2492	0
N_2	0	0	0	0	0	1	1	0	1.1191	0.7538	1
N_3	0	0	0	1	0	1	1	0	-0.7898	0.3122	0
N_4	0	0	0	1	1	1	1	0	0.6025	0.6462	1
N_5	0	0	1	1	0	0	0	0	5.9900	0.9975	1

N_6	0	1	0	1	0	0	0	0	-1.4840	0.1848	0
N_7	1	0	0	0	0	0	0	0	6.6357	0.9987	1
N_8	0	0	0	0	0	1	1	1	-2.6600	0.0654	0
N_9	0	0	1	0	1	1	1	1	-10.2696	3.4672e-05	0
N_{10}	1	0	0	1	0	0	0	0	3.8676	0.9795	1

На основі результатів досліджень та проведених експериментів був створений додаток, який виконує пошук в Інтернеті ресурсів з проху-адресами та перевіряє їх на працездатність. В основі додатку використувалася мова програмування PHP та фреймворк Symfony. Знайдені проху-адреси зберігаються в базі даних, яка використовується в додатку (MySQL). Це пов'язано з тим, що фреймворк Symfony має вбудований функціонал для швидкого підключення і безпосередньої роботи з реляційними базами даних. Даний додаток було реалізовано як команду, яку можна запускати в реальному часі і результати виконання якої заносяться в базу даних. Розроблений додаток може бути модифікований або інтегрований як сервіс в інші додатки.

Висновки

Проху-сервер залежно від потреб використання забезпечує різні рівні функціональності, безпеки та конфіденційності, і який можна налаштувати як веб-фільтр або брандмауер, що захищає комп'ютер від інтернет-загроз. При підключенні до проху-серверу необхідно визначати його працездатність, тобто можливість відправляти запити на різні ресурси або сервіси та отримувати відповіді (request/response).

Відзначені фактори, які можуть впливати на відправлення запиту, з використанням проху-адреси, та отримання відповіді від ресурсу, а також розглянуті умови для визначення працездатності проху-адресів. Було проаналізовано роботу проху-адресів, проху-серверів, та умови, при яких можна вважати проху-адресу працездатною, а при яких ні.

Для створення моделі оцінки працездатності проху-адресів було застосовано нейромережвий підхід. В якості моделі для оцінки працездатності проху-адреси використано НМ у вигляді тришарового перцептронів. Були сформовані набори навчальних

даних, що відповідають стану працездатності проху-адресів, котрі знаходяться у вільному доступі, і виконане навчання НМ методом зворотного розповсюдження помилки.

Створену НМ було перевірено на визначення стану працездатності проху-адресів за допомогою тестових наборів даних. Отримані результати показали здатність НМ виявляти працездатні та не працездатні проху-адреси.

На основі результатів досліджень та проведених експериментів був створений додаток, який виконує пошук в Інтернеті ресурсів з проху-адресами, перевіряє їх на працездатність та може бути модифікований або інтегрований як сервіс в інші додатки.

Список літератури:

- [1] What is a Proxy Server? How does it work? URL: <https://www.fortinet.com/resources/cyberglossary/proxy-server>.
- [2] Adrian R. Compare Different Types of Proxies. Best Proxy Reviews. URL: <https://www.bestproxyreviews.com/different-types-of-proxies/#types-of-proxies-based-on-ip-origin>.
- [3] Anonymous network proxy server. Proxy security or Should I trust Public Proxy Servers. URL: <https://comuedu.ru/en/windows/anonymous-network-proxy-server-proxy-security-or-should-i-trust-public-proxy-servers.html>.
- [4] Transparent proxy. URL: <https://www.imperva.com/learn/ddos/transparent-proxy/>.
- [5] What is an Anonymous Proxy and How Can It Benefit You? URL: <https://proxyway.com/guides/what-is-anonymous-proxy>.
- [6] The Different Pros and Cons of a Proxy Server URL: <https://securityonline.info/the-different-pros-and-cons-of-a-proxy-server/>.
- [7] How to check if the proxy is operable? URL: <https://socproxy.ru/blog/post/proverka-rabotosposobnosti-proksi>.
- [8] Online proxy server verification URL: <https://hidemy.name/en/proxy-checker/>.
- [9] Implementing health checks by David Yanacek. URL: <https://aws.amazon.com/builders-library/implementing-health-checks/>.
- [10] Timeouts, retries, and backoffs with jitter by Marc Brooker. URL: <https://aws.amazon.com/builders-library/timeouts-retries-and-backoff-with-jitter/>.
- [11] Stuart J. Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 4th US Edition, 2022. — 1151 p.
- [12] Khaikin S. Neural networks. Full course: A Comprehensive Foundation, 2nd edn, Williams, Moscow, 2006. — 1104 p.

Надійшла до редколегії 24.05.2022

УДК691.32/.34-033.33:51-047.58.

DOI 10.30837/bi.2022.1(98).04

Я.О. Ляшок¹, С.В. Подкопаєв², О.І. Повзун³, С.О. Вірич⁴, В.В. Калиниченко⁵¹Доктор економічних наук, професор, ректор, ДонНТУ, м. Луцьк, Україна
ORCID iD: 0000-0002-7643-8485²Доктор технічних наук, професор, проректор з наукової роботи,
ДонНТУ, м. Луцьк, Україна³Кандидат технічних наук, доцент, ДонНТУ, м. Луцьк, Україна
ORCID iD: 0000-0001-8183-9585, povzun.aleksey@gmail.com⁴Кандидат технічних наук, доцент, завідувачка кафедри, ДонНТУ, м. Луцьк, Україна⁵Кандидат технічних наук, доцент, завідувач кафедри, ДонНТУ, м. Луцьк, Україна

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТЕХНОЛОГІЧНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ СИМПЛЕКС-ПЛАНУВАННЯ ЕКСПЕРИМЕНТУ

Проведено математичне й комп'ютерне дослідження технологічної системи – об'ємногідрофобізований ніздрюватий бетон автоклавного тверднення з кремнійорганічною поліметилсилоксановою добавкою (ПМС-100). За допомогою симплекс-планування експерименту показано, що взаємозв'язок між складом сировинної суміші, технологічними факторами, структурою та фізико-механічними властивостями ніздрюватого бетону описується моделями у вигляді планів другого порядку на гіперкубі відповідної розмірності. Для визначення дисперсії відтворюваності виконано триразове повторення дослідів при значеннях вхідних параметрів на нульових рівнях.

МОДЕЛЬ МАТЕМАТИЧНА, СИМПЛЕКС-ПЛАНУВАННЯ ЕКСПЕРИМЕНТУ, ФАКТОР ВАРІЮВАННЯ, ПАРАМЕТР ОПТИМІЗАЦІЇ, БЕТОН

Yaroslav Liashok, Serhii Podkopaiev, Oleksii Povzun, Svitlana Virych, Valerii Kalynychenko. Mathematical modeling of a technological system using simplex planning of the experiment. Mathematical and computer research for optimization of the composition of bulk hydrophobized cellular concrete of autoclave hardening with silicone polymethylsiloxane additive PMS-100. Using simplex planning of the experiment it is shown that the relationship between the composition of the raw mixture, technological factors, structure and physical and mechanical properties of aerated concrete is described by models in the form of second-order plans on a hypercube of appropriate dimension. To determine the reproducibility variance, the experiments were repeated three times at the values of the input parameters at zero levels.

MATHEMATICAL MODEL, SIMPLEX EXPERIMENTAL PLANNING, VARIABLE FACTOR, OPTIMIZATION PARAMETER, CONCRETE

Вступ

Перспективним теплоефективним будівельним матеріалом є ніздрюватий бетон, частка виробів з якого в загальному обсязі стінових матеріалів нині становить 6-8%. За прогнозними даними обсягів розвитку житлового будівництва річна потреба у ніздрюватобетонних виробках зростає в 15 разів порівняно з існуючими обсягами його виробництва. Це пов'язано зі світовою тенденцією підвищення опору теплопередачі стінових конструкцій.

Розроблення нових технологій виробництва виробів з ніздрюватих бетонів дозволить виготовляти вироби, при застосуванні яких зовнішні огорожувальні конструкції будуть відповідати нормативному значенню опору теплопередачі — 1,7-2,2 м²·°C/Вт для різних кліматичних зон.

1. Постановка задачі

Необхідність економії матеріальних та енергетичних ресурсів при виробництві ніздрюватого бетону вимагає чіткого керування технологією, властивостями матеріалу та якістю готової продукції в цілому. Технологічний процес, що містить сукупність тісно пов'язаних між собою факторів, є складною статистичною системою.

Ефективний аналіз та дослідження об'єктів такого роду можливі на основі системного підходу, що передбачає побудову узагальненої моделі, яка б відбивала взаємозв'язки між елементами системи [1].

Однак за умов великої кількості факторів, що впливають на властивості матеріалу, практично неможливо розробити модель, здатну ефективно функціонувати для будь-якої технології та будь-якої сировини. У такому випадку задача зводиться до моделювання процесу одержання бетону на основі цементу, вапняно-кремнеземистого в'язучого, кремнеземистого компонента (мелений кварцовий пісок) та кремнійорганічної добавки.

2. Аналіз досліджень і публікацій

Засновниками використання системного підходу до вивчення технології ніздрюватих бетонів є О.О. Федін, Є.М. Чернишов та В.О. Вознесенський [2-6]. В роботі [5] детально наведено приклади застосування двофакторного ортогонального квадратного плану, двофакторного плану 3² для побудови неповних кубічних моделей, трифакторного некомпозиційного плану типу Бокса-Бенкіна, двофакторного симплекс-сумуючого плану (на прикладі введення в суміш мікродобавок — регуляторів якості

матеріалів для заощадження дефіцитного компонента), техніко-економічну інтерпретацію моделі і оцінювання впливу факторів на вихід (на прикладі для ніздрюватих композитів) тощо.

Для єдиного простору трьох рецептурно-технологічних факторів (РТФ) отримано 24 експериментально-статистичні моделі, що створюють два порівнювані комплекси: «структурні характеристики — РТФ» і «технічні властивості — РТФ», які відбивають закономірності впливу РТФ (складу ніздрюватобетонної суміші, виду газоутворювача і тривалості ізотермічної витримки) на характеристики структури і властивості матеріалу [7].

Запропонована розрахункова модель зводиться до розв'язання системи рівнянь, в яких міцність на стиск ніздрюватого бетону виражена як відношення максимального навантаження до площі елементарної несучої комірки. На підставі достовірної залежності міцності на стиск і її графічних ізоліній, одержаних при математичній обробці результатів експерименту, доведено, що визначальним фактором оптимізації є співвідношення між товщиною міжпорової перегородки і діаметром пор, яке для різних видів ніздрюватого бетону може становити 0,1-1,0. Моделювання свідчить про наявність у ніздрюватому бетоні псевдоніздрюватої крупнокапілярної, ніздрюватої капілярної однорідної або крупнопорової неоднорідної структури.

В роботі [9] за методом ортогонального центрального композиційного планування (з побудовою відповідних номограм) проведено чотирифакторний експеримент. Після статистичної комп'ютерної обробки експериментальних даних були одержані поліноміальні моделі другого порядку зміни міцності на стиск, середньої густини та коефіцієнта розм'якшення силікатних зразків. Виявлено оптимальні склади будівельних матеріалів автоклавного тверднення, проведено дослідження впливу окремих компонентів і режимів автоклавної обробки на їх технологічні та фізико-механічні властивості [9].

3. Формулювання цілі та завдань дослідження

Теоретичні узагальнення і прикладні розв'язання питань науково-інженерної проблеми управління технологічними процесами і якістю будівельних матеріалів становлять нині одне з найактуальніших напрямків досліджень. З розробленням цього напрямку пов'язані умови і перспективи «наукового володіння виробництвом» і можливості переходу виробництва на принципи оптимальності хіміко-технологічних рішень за критеріями матеріало- та енергоємності одержання будівельних матеріалів, виробів, конструкцій. По суті йдеться про створення цілісної системи управління технологією і якістю продукції, що являє собою багатопланову задачу, центральною частиною якої є розкриття фізико-механічної, механо-хімічної сутності формування якості продукції шляхом її оптимізації.

Метою роботи є за допомогою математичного моделювання і оптимізації залежності зміни фізико-механічних властивостей від технологічних параметрів автоклавного тверднення одержання високоефективного ресурсоекономічного об'ємногідрофобізованого ніздрюватого бетону.

Використовуючи комп'ютерні програми оптимізаційного конструювання будівельних композитів, завданнями дослідження є:

1. Розробити математичну модель
2. Побудувати номограми залежностей параметрів оптимізації системи від факторів варіювання
3. Визначити оптимальний склад ніздрюватого бетону

4. Викладення основного матеріалу досліджень

Як об'єкти дослідження прийнято:

- кремнеземистий компонент – мелений кварцовий пісок;
- портландцемент марки М500;
- вапно комове негашене за ДСТУ Б В.2.7-90:2011;
- кремнійорганічна добавка – поліметилсилоксанова (ПМС-100);
- вода за ДСТУ Б В.2.7-273:2011;
- газоутворювач – пудра алюмінієва пігментна (ПАП-1).

Для вивчення взаємозв'язку між технологічними факторами, структурою та властивостями ніздрюватого бетону автоклавного тверднення було застосовано симплекс-планування експерименту. Цей взаємозв'язок описується моделями у вигляді планів другого порядку на гіперкубі відповідної розмірності [5].

Відповідно до алгоритму симплексного методу [10] експериментальне визначення оптимальних значень цільової функції здійснювали за такою послідовністю:

1) обирали цільову функцію, інтервали варіювання, інтервали факторів варіювання і виконували кодування факторів;

2) уточнювали план експерименту, тобто обчислювали координати початкового симплекса. Кількість дослідів $N = k + 1$;

3) проводили відповідні досліди за планом експерименту;

4) упорядковували відгуки згідно зі ступенем їх зменшення і виділяли найменше значення відгуку Y_{imin} , яке потім відкидали;

5) будували новий симплекс. Він утворюється вершинами, що залишилися, вихідного симплекса і новою вершиною, яка виходить шляхом дзеркального відбиття відкинутої відносно протилежної грані вихідного симплекса. Координати нової точки x обчислювали за формулою:

$$x^* = \frac{2}{k} (x^1 + x^2 + \dots + x^{i-1} + x^{i+1} + \dots + x^{k+1}) - x^i = \frac{2}{k} \sum_{i=1}^{k+1} x^i - \left(\frac{2}{k} + 1 \right) x^i \quad (1)$$

де x^i – координати вершини вихідного симплекса з найменшим значенням цільової функції.

Побудова плану у вигляді симплекса дозволяє визначити (передбачити) відгук у новій вершині x^* :

$$\hat{y}^* = \frac{2}{k} \sum_{i=1}^{k+1} \hat{y}^i - \left(\frac{2}{k} + 1 \right) \hat{y}^i \quad (2)$$

6) проводили експеримент у новій точці x^* і одержували експериментальне значення y^* цільової функції;

7) послідовно переміщували симплекс, під час чого на кожному кроці відбувалось скидання вершини симплекса з найменшим значенням цільової функції і реалізація досліду у новій вершині. При цьому напрямок руху співпадає з направленням градієнта.

Рівні варіювання технологічних факторів призначали на основі попередньо проведених однофакторних експериментів. Результати експериментів обробляли методами математичної статистики [11], [12]. Побудову поліноміальних моделей здійснювали за методом повного перебору, що дозволяло серед множини можливих (при заданій структурі) обрати найкращу модель за критерієм Фішера. Довірчі інтервали оцінок отриманих результатів визначали на основі результатів випробувань малих вибірок із застосуванням розподілу Стюдента при надійності, що дорівнювала 0,95. Для обчислення дисперсії відтворюваності виконано триразове повторення дослідів при значеннях вхідних параметрів на нульових рівнях. Однорідність дисперсій перевіряли за критерієм Кохрена. Адекватність моделі перевіряли за критерієм Фішера.

Для систем, що є сумішами різних компонентів, змінні x_i ($i=1, 2, 3, \dots, n$) є пропорціями (відносним вмістом) компонентів суміші та задовольняють умові:

$$\sum_{1 \leq i \leq n} x_i = 1 \quad (3)$$

Геометричне місце точок, що задовольняють умові нормованості суми змінних (3), являє собою $(n-1)$ -мірний правильний симплекс (трикутник для $n=3$, тетраедр для $n=4$ тощо). Кожній точці такого симплекса відповідає суміш визначеного складу і, навпаки, будь-якій комбінації відносних вмістів компонентів відповідає визначена точка симплекса.

Координати компонентів визначали у симплексній системі координат, в якій відносні вмісти кожного компонента відкладаються вздовж відповідних сторін (граней) симплекса від 0 до 1. Отримані апроксимуючі моделі можуть бути використані для передбачення відгуків в будь-якій точці симплексу. Точність передбачення відгуку в різних точках симплексу різна і окрім координат точки залежить від екстремальної ситуації (дисперсії досліду, кількості паралельних спостережень у вузлах симплексної ґратки).

Для підбору та моделювання рецептур ніздрюватобетонної суміші проведено підготовку сировинних компонентів, що включала здрібнення і усереднення складових, підігрів води і приготування водно-алюмінієвої суспензії.

Вплив широкого діапазону концентрацій кремнійорганічної поліметилсилоксанової добавки ПМС-100 на міцність ніздрюватого бетону на стиск визначали на зразках-кубах $100 \times 100 \times 100$ мм. Зразки були виготовлені згідно з вимогами [13] з бетонної суміші рецептури (Цемент : Вапно : Пісок) $1 : 0,61 : 0,89$. Рухомість суміші за Суттардом [14], [15] – 14 ± 2 см. Зразки витримували в формах до автоклавної обробки впродовж 12 годин, після чого запарювали в лабораторному автоклаві місткістю $0,07 \text{ м}^3$ за режимом: прогрів та підйом тиску пари до 1 МПа – 3 години; витримування при тиску пари 1 МПа – 6 годин; зниження тиску пари за рахунок природного охолодження.

З метою моделювання складу ніздрюватобетонної суміші (табл. 1) проведено аналіз процесів синтезу гідросилікатів з використанням розробленої теоретичної моделі. Кремнійорганічну поліметилсилоксанову добавку ПМС-100 приймаємо в розрахунок як високоактивний кремнеземистий компонент, вважаючи, що присутність органічної складової добавки не впливає суттєво на склад новоутворень. Коефіцієнт гідралічної активності добавки ПМС-100 приймемо 0,5, а для піску 0,035.

Таблиця 1

Витрата компонентів ніздрюватобетонної суміші для виготовлення бетону густиною 400 кг/м^3

Витрата компонентів на 1 м^3 , кг					
вапно	цемент	кремнеземистий компонент	вода	добавка	алюмінієва пудра
95,28	155,64	138,17	155	7,8	0,55

Збільшення вмісту добавки ПМС-100 понад 2% спричиняє зниження міцності готових виробів. Однією з можливих причин такого явища є понадоптимальна автоклавна обробка (тобто, починаючи з 10-ї години новоутворення перекристалізуються та укрупнюються, що викликає порушення структури). Іншою – утворення лінз гідрофобізатора, що спричиняють розклинюючу дію та викликають порушення зв'язків між новоутвореннями.

Досліджувана добавка ПМС-100 фактично реагує вже на стадії змішування, формування та доавтоклавної витримки. Тому доречним стало розглянути розрахунок синтезу за умови, що добавка на початок синтезу (тобто в 0 годин) повністю переходить у новоутворення. Оптимальні склади високоефективного ресурсоекономічного об'ємногідрофобізованого ніздрюватого бетону визначали як оптимальні області допустимих значень факторів варіювання X_1, X_2, X_3 (табл. 2).

Таблиця 2

Значення факторів варіювання

Система		Фізичний зміст фактора варіювання		
		Вміст портланд-цементу, кг	Вміст вапна, кг	Вміст добавки ПМС-100, кг
		X_1	X_2	X_3
Ніздрюватий бетон	Інтервал варіювання	10	20	1
	Рівні фактора у кодіваних числах	-1	0	+1
		120	130	140
		40	60	80
		1	2	3

Ці оптимальні області обмежені поверхнями рівня функції відгуку за кожним з параметрів оптимізації (табл. 3).

Матриця планування експерименту і отримані значення параметрів оптимізації надані в табл. 4.

За критерій оптимальності плану прийнято критерій D -оптимальності, який пов'язаний з мінімізацією об'єму еліпсоїду розсіювання оцінок параметрів рівнянь регресії [16]. Для створеного плану експерименту за методом найменших квадратів обчислено коефіцієнти рівнянь регресії за виразом [17]:

$$b_i = (X^* \cdot X)^{-1} \cdot X^* \cdot Y,$$

де b_i – коефіцієнт рівняння регресії; X – матриця плану експерименту; X^* – транспонована матриця плану експерименту; $(X^* \cdot X)^{-1}$ – матриця, обернена до добутку матриці плану експерименту на його транспоновану матрицю; Y – вектор-стовпець результатів експерименту.

За планом з однією центральною точкою досліджено 15 сумішей (табл. 4).

Таблиця 3

Параметри оптимізації системи

№ з.п.	Код параметра оптимізації	Фізичний зміст параметра оптимізації	Граничні значення функції відгуку
1	Y_1	Міцність на стиск бетону, $\sigma_{ст}$, МПа	2,90; 3,27
2	Y_2	Морозостійкість бетону, F , цикли	> 25; > 25
3	Y_3	Водопоглинання бетону, W , %	<8; < 8
4	Y_4	Середня густина бетону, ρ_0 , кг/м ³	351; 449

Сировинну суміш готували шляхом спільного перемішування цементу, вапняно-кремнеземистого в'язучого, кварцового піску, води, добавки ПМС-100 та водної суспензії алюмінієвої пудри (ПАП-1) та поверхнево-активної речовини (ПАР).

Вапняно-кремнеземисте в'язуче готували спільним помелом вапна та піску у співвідношенні 1:1,8 до питомої поверхні 550 м²/кг. Активність вапняно-кремнеземистого в'язучого становила 44%.

Суспензію газоутворювача готували при інтенсивному перемішуванні алюмінієвої пудри ПАП-1 з водою при температурі 60-65 °С. Витрата ПАП-1 від 0,2% до 0,6% від маси сухих компонентів.

У всіх точках плану прийнято витрату води, що забезпечує рухомість суміші на віскозиметрі Суттарда 16 см. При цьому водотверде відношення змінювалося у межах 0,4-0,5.

Обробка результатів експерименту і обчислення коефіцієнтів рівнянь регресії з урахуванням їх значущості дозволили одержати поліноміальні моделі.

Статистичний аналіз отриманих результатів містить перевірку двох статистичних гіпотез:

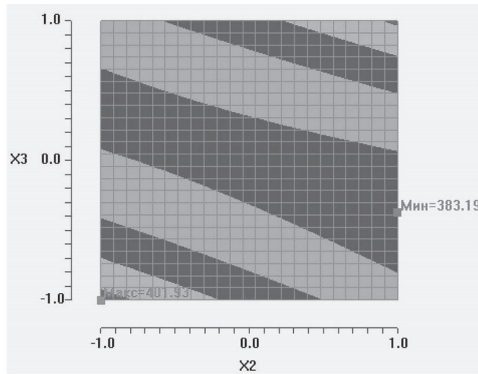
Таблиця 4

Матриця планування та результати експерименту

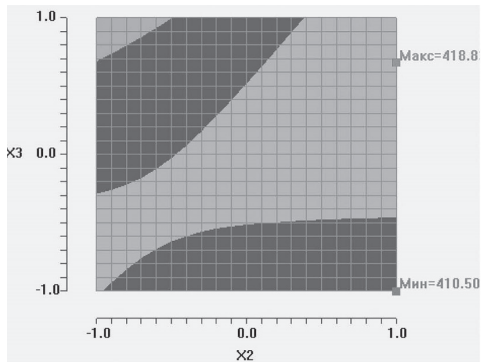
№	План			Результат							
	X_1	X_2	X_3	Y_1		Y_2		Y_3		Y_4	
				Y_{11}	Y_{12}	Y_{21}	Y_{22}	Y_{31}	Y_{32}	Y_{41}	Y_{42}
1	2	3	4	5	6	7	8	9	10	11	12
1	-1	-1	-1	20,8	26,1	19	21	5,0	5,8	412	420
2	-1	-1	+1	26,2	28,1	20	22	0,8	1,2	380	392
3	-1	+1	-1	22,0	24,1	20	24	4,9	5,6	386	408
4	+1	-1	-1	21,8	23,6	20	22	5,1	5,8	404	412
5	-1	+1	+1	24,6	26,2	20	22	0,8	1,4	410	418
6	+1	-1	+1	21,0	22,8	19	20	1,0	1,3	398	420
7	+1	+1	-1	22,6	26,8	21	23	4,6	4,8	402	414
8	+1	+1	+1	29,1	32,1	21	24	0,8	1,2	396	416
9	+1	0	0	34,1	36,2	26	30	1,2	1,9	380	410
10	-1	0	0	36,1	38,2	24	26	1,9	2,2	390	420
11	0	+1	0	32,0	34,0	25	30	1,4	2,1	380	420
12	0	-1	0	31,8	35,1	27	31	1,7	2,2	385	415
13	0	0	+1	28,1	32,1	22	26	1,0	1,3	402	412
14	0	0	-1	27,4	32,6	20	21	4,2	5,0	380	416
15	0	0	0	32,6	34,1	24	28	1,6	2,4	394	414
Норматив				28,97	32,74	>25	>25	<8	<8	351	449

- про значущість отриманих коефіцієнтів моделі;
- про адекватність представлення результатів експерименту здобутим рівнянням регресії.

Результати розрахунку математичної моделі з подальшим регресійним аналізом наведено на рис. 1 – рис. 4.

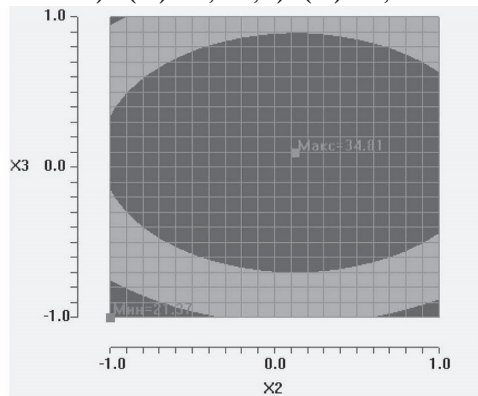


а

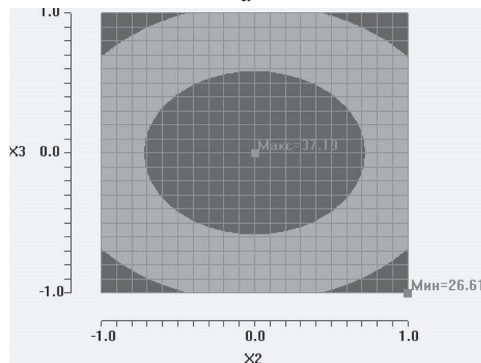


б

Рис. 1. Номограма залежності функції відгуку (середня густина ніздрюватого бетону ρ_0) від вмісту вапна і вмісту добавки ПМС-100 а) – $(tS)e=2,517$; б) – $(tS)e=6,0$

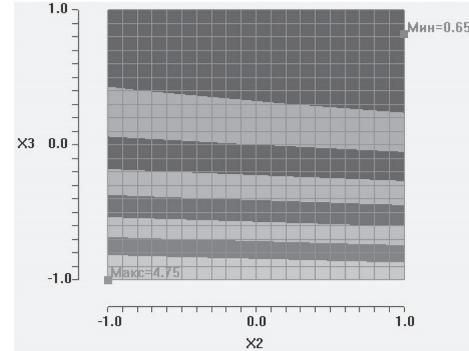


а

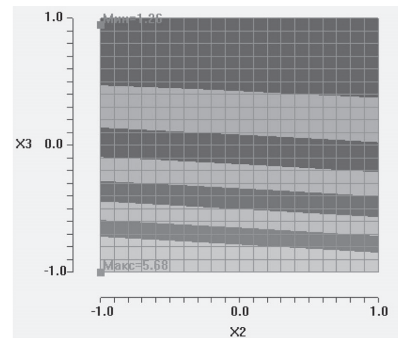


б

Рис. 2. Номограма залежності функції відгуку (міцність ніздрюватого бетону на тиск $\sigma_{ст}$) від вмісту вапна і вмісту добавки ПМС-100 а) – $(tS)e=2,517$; б) – $(tS)e=2,79$

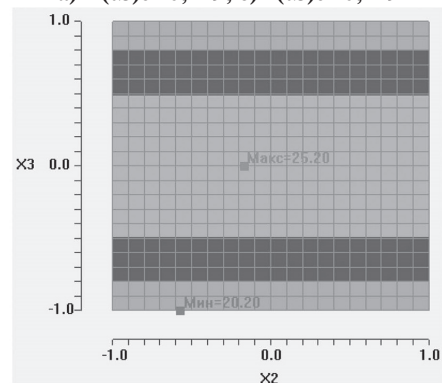


а

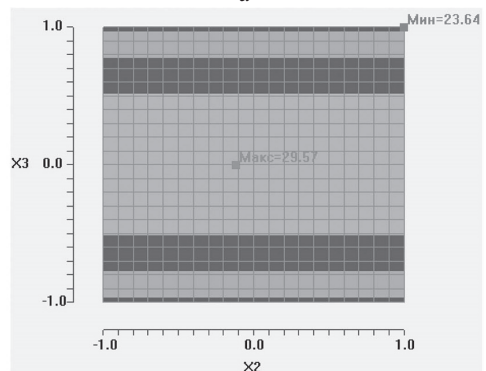


б

Рис. 3. Номограма залежності функції відгуку (водопоглинання ніздрюватого бетону) від вмісту вапна і вмісту добавки ПМС-100 а) – $(tS)e=0,229$; б) – $(tS)e=0,279$



а



б

Рис. 4. Номограма залежності функції відгуку (морозостійкість ніздрюватого бетону F) від вмісту вапна і вмісту добавки ПМС-100 а) – $(tS)e=2,027$; б) – $(tS)e=2,280$

На основі аналізу результатів експерименту встановлено оптимальний склад сировинної суміші для автоклавного об'ємногідрофобізованого ніздрювато-го бетону, що забезпечує підвищені фізико-механічні характеристики (табл. 5).

Таблиця 5

Оптимальний склад ніздрювато-го бетону

Компонент	Вміст, % від маси сухих
Портландцемент	140 кг (35%)
Вапно	60 кг (15%)
Пісок кварцовий	189 кг (47,25%)
Добавка ПМС-100	2%
Газоутворювач	0,55%
Поверхнево-активна речовина	0,2%

Висновки та напрямок подальших досліджень

На підставі розробленої математичної моделі за допомогою симплекс-планування експерименту ніздрюватий бетон оптимального складу характеризується підвищеною якістю.

За вмісту 2% (від маси сухих компонентів) кремнійорганічної поліметилсилоксанової добавки (ПМС-100) фізико-механічні властивості досліджуваного ніздрювато-го бетону є такими:

- середня густина – 400 кг/м³;
- міцність на стиск – 2,9 МПа;
- морозостійкість – 25 циклів поперемінного заморожування–відтавання;
- водопоглинання – < 5%.

Подальші роботи будуть присвячені впровадженню результатів наукових досліджень у виробництво (випуск дослідно-промислової партії об'ємногідрофобізованого ніздрювато-го бетону автоклавного тверднення).

Список літератури:

- [1] Коверніченко Л.М., Хільченко О.П. Штучні будівельні матеріали // Вісник Криворізького національного університету. – 2019. – Вип. 48. – С. 111-114.
- [2] Федін А.А., Чернышов Е.М. Влияние состава сырьевой шихты на структуру цементирующих веществ и физико-механические свойства силикатного ячеистого бетона // Исследования по цементным и силикатным бетонам. – Воронеж: ВГУ, 1970 – Вып. 4. – С.15-29.
- [3] Чернышов Е.М. Статистический анализ структуры силикатных автоклавных материалов и его приложения к изучению свойств, определяющих стойкость // Долговечность конструкций из автоклавных бетонов. – Таллин: Валгус, 1981. – Ч.1. – С. 14-18.
- [4] Методические рекомендации по составлению технического задания на моделирование рецептурно-технологических свойств строительных материалов и изделий / В.А. Вознесенский, И.Б. Удачкин, Б.Л. Огарков и др. – К.: 1981. – 57 с.
- [5] Вознесенский В.А. Статистические методы планирования эксперимента в технико-экономических исследованиях 2-е изд. – М.: Финансы и статистика, 1981. – 263 с.
- [6] Вознесенский В.А. Оптимизация состава многокомпонентных добавок и композитов. – К.: Знание, 1981. – 20 с.
- [7] Шинкевич Е.С. Оптимизация структуры ячеистого силикатного бетона по комплексу критериев качества на основе изопараметрического анализа: дис. к-та техн. наук: – Елена Святославовна Шинкевич. – Воронеж. – 1985. – 239 с.
- [8] Королев А.С. Оптимизация состава и структуры конструкционно-теплоизоляционного ячеистого бетона / А.С. Королев, Е.А. Волошин, Б.Я. Трофимов // Строительные материалы. – 2004. – № 4. – С. 30-32.
- [9] Володченко А.Н. Оптимизация свойств силикатных материалов на основе известково-песчано-глинистого вяжущего / А.Н. Володченко, Р.В. Жуков, В.С. Лесовик, Е.А. Дороганов // Строительные материалы. – 2007. – №4. – С. 66-68.
- [10] Кононюк Е.А. Основы научных исследований (Общая теория эксперимента) / Е.А. Кононюк, Книга 3. – Монография. – К.: Освіта України. – 2011. – 456 с.
- [11] Кожухова Н.И. Теоретические основы планирования, проведения и обработки эксперимента / Н.И. Кожухова. – 2019. – 76 с.
- [12] Солодов В.С. Практикум по планированию, проведению и обработке эксперимента в исследовании технологических процессов / В.С. Солодов. – 2018. – 150 с.
- [13] ДСТУ-Н Б В.2.7-308:2015. Настанова з виготовлення виробів з ніздрювато-го бетону. [Чинний від 01.10.2016]. Київ: Мінбуд України, 2016. 52 с.
- [14] Дворкін Л.Й., Житковський В.В. Технологія опоряджувальних, теплоізоляційних та гідроізоляційних матеріалів: навч. посіб / Л.Й. Дворкін, В.В. Житковський. – Рівне: НУВГП. – 2010. – 227 с.
- [15] ДСТУ Б В. 2.7 - 82 :2010. В'язучі гіпсові. Технічні умови. [Чинний від 01.03.2011]. Київ: Мінбуд України. – 2011. – 29 с.
- [16] Голикова Т.И. Свойства D-оптимальных планов и методы их построения / Т.И. Голикова, Н.Г. Микешина // Новые идеи в планировании эксперимента. – М.: 1969. – С. 34-39.
- [17] Кафаров В. В. Методы кибернетики в химии и химической технологии / В.В. Кафаров. – М.:Химия, 1971. – 496 с.

Надійшла до редколегії 21.06 2022

УДК 007.519.7

DOI 10.30837/bi.2022.1(98).05

¹Г.Г. Четвериков, ²І.М. Сикал, ³І.С. Табакова

¹Доктор технічних наук, професор,
Харківський національний університет радіоелектроніки,
Україна, ORCID iD: 0000-0001-5293-5842, chetvergg@gmail.com

²Кандидат медичних наук, доцент кафедри акушерства та гінекології №2, лікар УЗД вищої категорії,
Харківська медична академія післядипломної освіти МОЗ України,
iryna.sykal@gmail.com

³Кандидат технічних наук, доцент кафедри медіасистем та технологій,
Харківський національний університет радіоелектроніки,
Україна, ORCID ID: 0000-0001-6629-4927, iryna.tabakova@nure.ua

ДЕЯКІ АСПЕКТИ МОДЕЛЮВАННЯ ПРИРОДНОМОВНИХ ПРОЦЕСІВ ТА ЇХ ЗАСТОСУВАННЯ ПРИ ВИКЛАДАННІ В ВИЩИХ УЧБОВИХ ЗАКЛАДАХ

Стаття присвячена аналізу проблеми створення систем штучного інтелекту, які дозволяють моделювати на логічному та апаратному рівнях лінгвістичні відношення що описані алгебро-логічними операціями над природною мовою, і які є елементами k -значної структурної організації інформаційно-інтелектуальних технологій: АСП-структури.

ЛОГІКА, ПРИРОДНА МОВА, ШТУЧНИЙ ІНТЕЛЕКТ, k -ЗНАЧНА СТРУКТУРА, АСП-СТРУКТУРА

Вступ

У XXI столітті виникла матеріально-енергетична та соціально-економічна криза суспільного виробництва, виходом із якої став перехід економічно розвинутих країн від матеріально-енергетичних технологій виробництва та споживання до інформаційно-інтелектуальних технологій, де економія матеріально-енергетичних ресурсів досягається шляхом переходу від опрацювання інформації до обробки знань. Інтелектуальна підтримка суспільно-економічних процесів дозволяє раціонально розв'язати триєдине завдання прогресу сучасної цивілізації – підвищення ефективності, гуманістичності та екологічності виробництва [1-3].

Суспільна ефективність в умовах переходу до соціально орієнтованих ринкових відносин принципово залежить від якості прийняття управлінських рішень. Це надає актуальності проблемам створення систем штучного інтелекту, які дозволяють формалізувати у вигляді знань інтереси суб'єктів господарювання, моделювати особистісний фактор при прийнятті економічних рішень. Проте проектування та впровадження в практику таких систем потребує не тільки розв'язання теоретичних проблем у галузі подання, поповнення, оброблення та збереження знань стосовно програмної їх реалізації, а й досліджень у галузі створення нових структурних і елементних рішень, розробки теорії та методів ефективного кодування. Найшвидші суперкомп'ютери сьогодні працюють набагато повільніше, ніж це потрібно для роботи з даними і знаннями, поданими мовними моделями, що описують описові науки. Розв'язання цієї проблеми залежить від того, чи будуть створені відповідні комп'ютерні системи, здатні працювати зі швидкістю у 1000 разів більшою, аніж існуючі найбільші

суперкомп'ютери [1-3]. Поняття „нові інформаційні технології” передбачає вміння користуватися сучасними засобами та методами добування, опрацювання і систематизації знань.

Сучасна система штучного інтелекту (ШІ) (ЕОМ 5-го покоління) передбачає нові додаткові компоненти, не властиві традиційним архітектурам на базі Нойманівських процесорів, а саме – лінгвістичного процесора (транслятора), вирішувача задач та бази знань, координація дій яких реалізується з допомогою центрального процесора.

Очевидно, що єдиним відомим нам, об'єктивним носієм знання та інтелекту є людина, а виразником, засобом до зовнішнього спілкування та носієм інтелекту є людська мова.

Проблеми подання знань і роботу з ними зазвичай відносять до галузі інформаційних чи інтелектуальних технологій, виходячи з математико-програмістського підходу до постановки цього питання. Ми ж пропонуємо “природніший” розгляд проблеми подання з позиції відкритих систем і знань, та інтелекту як процесу.

1. Мета роботи

Основним завданням цієї роботи є виклад новостворюваної концепції організації інформаційно-інтелектуальних технологій та інтелектуальної підтримки суспільно-економічних процесів, зокрема діяльності фінансово-кредитних установ. Предметом досліджень є моделювання інтелектуальної діяльності людей як у зовнішньому її прояві (вирішення складних завдань, розуміння природної мови, інтерпретація візуальної інформації та мови), так і у внутрішньому (накопичення, надання і використання знань).

2. Основні поняття та визначення

У теорії пізнання постулюються для новостворюваних теорій два основоположних принципи: «якщо існує об'єкт, то його можна досліджувати» та «щоб уникнути труднощів взаєморозуміння, необхідно перш за все порозумітись із основними визначеннями», які стосуються даної теорії. Відповідно, *наочною чи проблемною галуззю* ми називатимемо сукупність даних і знань, що представляється достатнім для інформаційного (заданого на контекстно-залежній мові) забезпечення рішення деякої задачі чи сукупності завдань інтелектуальної системи.

Складність поняття *знань*, завжди виразимо на деякій мові відношень, полягає в множинності можливостей її реалізації і нерозривності з поняттям дані, *в безперервному процесі їх зміни, врахування якого забезпечує реалізацію контекстного зв'язку даних.*

Відзначимо сферу предметних галузей, де найдодільніше працювати з даними і знаннями, поданими мовними моделями. Це галузі з переважанням емпіричного знань, де складність фактів і їх описів виключає використання мови математики — так звані описові науки, які, загалом, і стали такими, бо вивчають те, що не може бути адекватно описане мовою математики — відкриті системи в їх природному оточенні, системи, спостережувані на рівні апарату лінгвістики. Як буде видно з подальшого, саме такий підхід дозволяє добиватися найбільшого ступеня адекватності опису тієї чи іншої системи, особливо при введенні поняття зміни формально-логічних властивостей лінгвістичного апарату під дією вхідних інформаційних потоків.

Сказане вище означає, що сукупність знання—дані в завданнях управління подається деякою семіотичною системою. Взагалі кажучи, під семіотикою розуміється наука про знаки і знакові системи, а також про мови, як знакові системи, причому загальний підхід до мовних знакових систем, що включає не тільки традиційне поняття мови, але і мови запахів, свисту, абстрактного живопису і тому подібне, вимагає ще свого теоретичного обґрунтування. Мовою може бути все, що забезпечує виникнення і існування комунікативної функції для своїх користувачів.

У семіотичній системі зазвичай виділяють три аспекти: *синтаксичний, семантичний і прагматичний.* Такого роду розділення цікавить нас із наступної причини. Відповідно до перерахованих трьох аспектів семіотичних систем необхідно виділити *три типи знань, як три типи відношень між даними:* синтаксичні, семантичні і прагматичні.

Знання синтаксичного типу характеризує синтаксичну структуру потоку інформації, яка не залежить від сенсу і змісту використовуваних при цьому понять, тобто інтелектуальну систему не утворює.

Семантичне знання розглядається як структура, утворююча поточний контекст. Воно містить інформацію, безпосередньо пов'язану з поточними

значеннями і сенсом описуваних понять і зумовлює стан зв'язків даних в інформаційній базі.

Прагматичне знання зумовлює найбільш вірогідні зв'язки, що описують дані з погляду вирішуваної задачі (узагальнений чи “об'єктивний” контекст), наприклад, із урахуванням специфічних критеріїв і угод, що діють у даному завданні.

Поняття об'єктивності відповідає трактуванню прагматики, прагматичного аспекту створення інтелектуальної системи, як спрямованого обмеження її “свободи волі”, обмеження непотрібних зв'язків і “фантазії зв'язків за контекстом” для системи, від якої ми не чекаємо надлишку інтелектуальності. Для порівняння пригадаємо, що ми маємо на увазі, коли даємо людині характеристику “прагматик”.

З інженерної точки зору, синтаксична, семантична і прагматична сторони знань є різними зв'язками одного чи групи термінів (даних) з іншими записами в інформаційній базі.

Зі сказаного зрозуміло, що найбільш загальна проблема побудови системи управління семантичного чи семантико-прагматичного рівня взаємодії, пов'язана з вибором технології контекстно-залежного подання знань, побудовою інформаційних баз (даних і знань) про предметну галузь і механізму висновку для отримання необхідних рішень.

Логіка цього механізму може мінятися залежно від контекстів взаємозв'язків знаків (їх прагматики), що є однією з найважливіших причин початкової організації всіх практичних робіт тільки на проблемно-орієнтованій підмножині контекстно-залежної мови, тобто там, де ми сподіваємося на деяку постійність контексту. Ця умова є корисною тільки в сенсі забезпечення деякої початкової простоти побудов. Але, на відміну від “чистих формально-логічних систем”, умова постійності контексту не є для нас обов'язковою. Всі наші побудови орієнтовані, кінець кінцем, саме на мінливості поточного контексту, без чого ми не маємо права говорити про інтелектуальні системи.

Структурування мови йде через формування базового набору відношень (їх число не більше 200): часових, просторових, казуальних, квантифікуючих (бінарні, тернарні) тощо. Метапроцедури — це цілеспрямований пошук у лабіринті можливостей, побудова чи пошук нових лабіринтів, структурування, декомпозиція задач на підзадачі, формування закономірностей на підставі спостережень. У основі наших знань про світ лежать системи класифікації, які дозволяють нам сприймати світ, як структурований за мірою загальності об'єкт.

3. Оброблення мов декларативного типу

Мови декларативного типу зручні для опису (подання) знань у системах ШІ [1-3]. Оброблення мови декларативного типу можна розбити на дві частини:

- те, що декларує людина;
- те, що опрацьовує ЕОМ.

Попереднє оброблення людиною полягає в абстрагуванні реальної проблеми в фізичну чи математичну модель. У подальшому математична модель перетворюється в машинно-орієнтоване відображення, яке сприймає ЕОМ. Складання відповідної програми для перетворення форм подання інформації можливе лише з допомогою семантичного опрацювання. ЕОМ при цьому здійснює лише підтримку оброблення, що його здійснює людина.

Основу такої машинної мови складають:

- основні поняття;
- семантика (на основі відповідного синтаксису);
- процедурні зображення, що описують семантичні одиниці;
- відповідні правила структурування понять (синтаксичні правила).

Механізми оброблення (їх робота) визначаються семантикою та синтаксисом мови [1-3]. Мови процедурного типу є мовами опису, що базуються на семантиці процедур. Операційні механізми цих мов відповідають функціям оброблення та перетворення інформації, а структурні елементи мови не мають нічого спільного з об'єктами реального світу. Процедурна семантика розглянута й описана в теорії обчислень, що дозволяє отримати повноцінний теоретичний опис системи оброблення знань у ШІ.

Мова декларативного типу опирається на відповідну їй структуру даних. При чому структура даних відповідає концептуальній структурі світу. Структура даних мови декларативного типу включає в себе компоненти, які визначаються структурою пам'яті ЕОМ. У мовах декларативного типу співвідношення синтаксис-семантика аналогічні співвідношенню природна мова — рисунки, графи, просторові зображення, бази даних тощо.

Бази даних — це абстрактні моделі обмеженої прикладної галузі (галузі знань), які поділяються на:

- a) інженерні БД;
- b) ієрархічні;
- c) мережеві;
- d) реляційні;
- e) постріляційні.

Подання моделей декларативних мов здійснюється предикатами, малюнками, кресленнями, графами тощо. Математична структура даних у декларативних мовах базується на системах предикатних рівнянь у алгебрі скінчених предикатів або на аксіоматичній теорії множин, у якій теорія множин інтерпретується як структура даних.

Таким чином, приходимо до можливо єдиного шляху, яким найповніше, найоб'єктивніше і найефективніше треба і можна йти шляхом створення систем ШІ — це шлях аналізу, моделювання та синтезу мовного інтелектуального інтерфейсу із допомогою АСП, засобів k -значної логічної системи та відповідних структур і кодування.

АСП [1] є узагальненням алгебри логіки і визначена як:

$$x^\delta = \begin{cases} x, & \delta = 1, \\ -x, & \delta = 0, \end{cases} \quad (1)$$

де δ — степінь аргументу.

Генеральною ідеєю АСП є розширення області пробігання δ до $\{a_0, a_1, \dots, a_{k-1}\}$, де a_0, a_1, \dots, a_{k-1} — k -значні змінні. Виходячи з цього, задаються функції розпізнавання символів a_i змінної x :

$$x^{a_i} = \begin{cases} 1, & x = a_i, \\ 0, & x \neq a_i. \end{cases} \quad (2)$$

При цьому фіксується множина змінних $x_1 \dots x_n$ та скінчених областей пробігання

$$A_j = \left\{ a_{i_1}, \dots, a_{i_{k_j}} \right\}$$

кожної змінної x_j , які можуть бути довільної значності.

Функції розпізнавання з'єднуються операціями $\&$, \vee булівської алгебри, у результаті чого отримуються вирази $x^a \& (y^b \vee x^c)$, названі в [1] скінченими предикатами. Вони трактуються як функції виду:

$$f: A_1 \times \dots \times A_n \rightarrow \{0, 1\}. \quad (3)$$

Це, по суті, характеристичні функції k -значної логіки, але, на відміну від неї, в АСП усі функції можуть приймати значення лише з множини $E_2 = \{0, 1\}$. Операції (1) розпізнавання, $\&$ та \vee утворюють функціонально повний базис у множині функцій виду (3).

Теорія інтелекту — це наука про детерміновані, дискретні і скінчені функції індивідуального людського інтелекту [1-3]. Таке визначення істотно обмежує поле діяльності для дослідника інтелекту.

По-перше, це визначення вимагає, щоби теорія інтелекту обмежилася вивченням лише функціональної сторони інтелекту, залишаючи осторонь структуру його матеріального носія. При цьому за межами теорії інтелекту залишаються такі цікаві і важливі питання, як будова мозку і нервової системи людини, функціонування їх окремих частин і структур.

По-друге, теорія інтелекту визначається як теорія індивідуального людського інтелекту. Отже, з теорії інтелекту виключаються процеси взаємодії людей один із одним, а це веде до винесення соціальних явищ за її межі. Крім того, таким визначенням виключаються з розгляду інші, ще не відкриті наукою, можливі види інтелекту, а також (взаємодія між інтелектом різних видів).

По-третє, прийняте тут визначення теорії інтелекту вимагає виключення з розгляду безперервних і нескінчених інформаційних процесів, веде до зменшення ролі чинника випадковості в діяльності інтелекту. Недостаток такого підходу приводить, зокрема, до неможливості усестороннього вивчення на його основі системи «інтелект — зовнішній світ».

Ігнорування статистичної природи сигналів, що надходять до інтелекту із зовнішнього світу, також істотно збіднює теорію інтелекту. Бажання обійтися без безперервних моделей сильно утрудняє вивчення роботи органів чуття, органів руху і мови людини.

Таким чином, прийняте нами визначення теорії інтелекту страждає багатьма недоліками, воно обмежує предмет цієї науки порівняно вузькими рамками, при цьому багато процесів і механізми, які люди схильні інтуїтивно відносити до розряду інтелектуальних, залишаються за межами цього визначення. Нижче приводяться міркування, що спонукали нас, не дивлячись на наголошені моменти, зупинитися на так недосконалому визначенні теорії інтелекту.

Перш за все, слід взяти до уваги ту обставину, що, навіть якщо б ми врахували всі ті доповнення, які тільки що були згадані, все одно повнота визначення теорії інтелекту не була б досягнута. Інтелект — це така система, яка знаходиться в постійному розвитку; ніхто не в змозі передбачати всі можливі напрями і наслідки цього розвитку. Не виключено також, що з часом будуть відкриті нові види інтелекту, принципово відмінні від людського. Тому визначення теорії інтелекту, як би ми не старалися, у будь-якому випадку буде неповним. З іншого боку, в майбутньому у зв'язку з успіхами пізнання інтелекту ніщо не перешкодить його дослідникам розширити рамки теорії інтелекту, коли в цьому з'явиться необхідність. В даний момент, проте, така необхідність гостро не відчувається. Навіть те, порівняно вузьке, визначення теорії інтелекту, яке прийняте нами, ставить масу цікавих задач; рішення цих задач і практичне використання досягнень теорії інтелекту обіцяє великі блага людству. В даний час, поки теорія інтелекту знаходиться ще у стадії зародження і робить перші кроки в своєму розвитку, вельми розширювальне тлумачення її предмету надало б їй «ведмедючу послугу», бо потурало б розпилюванню наукових сил і засобів, розтраті зусиль у безплідних спробах вирішувати задачі, складність яких перевершує можливості сучасної науки. І, навпаки, вдале тимчасове обмеження предмету теорії інтелекту дозволило б сконцентрувати зусилля на першочергових задачах теорії інтелекту, послужило б стимулом до якнайшвидшого її розвитку.

Визначаючи теорію інтелекту таким чином, як це зроблено в роботах [1-3], ми виходили з того погляду, що детермінованим, дискретним і скінченим процесам в інтелектуальній діяльності належить провідна роль. Вельми важливою обставиною, що вплинула на вибір такого визначення, послужила також наявність в даний час універсальних цифрових обчислювальних машин. Детерміновані, дискретні і скінчені процеси — це в точності ті процеси, які в принципі можуть бути реалізовані на цифрових обчислювальних машинах. Це надає в розпорядження теорії інтелекту універсальний засіб для практичної перевірки і апробації всіх її висновків і рекомендацій.

При прийнятому визначенні теорія інтелекту стає могутньою рушійною силою в справі подальшого вдосконалення засобів обчислювальної техніки. Це вдосконалення, у свою чергу, спричинить серйозну стимулюючу дію на розвиток самої теорії інтелекту.

Обмеження теорії інтелекту детермінованими, дискретними і скінченими процесами дає можливість забезпечити її універсальним математичним апаратом для формального опису довільних скінчених відношень. У роботі як такий апарат запропонована алгебра скінчених предикатів. На наш погляд, АСП може служити міцним фундаментом теорії інтелекту. Саме завдяки наявності цієї алгебри ми зважилися виділити теорію інтелекту в розряд самостійної наукової дисципліни.

Створення АСП доводить до логічного кінця пошуки універсальної математичної мови для опису детермінованих, дискретних і скінчених інформаційних процесів. АСП у відомому значенні є узагальненням апарату булевих функцій (перехід від двійкових знаків до букв) і апарату багатозначної логіки (перехід від функцій до відношень). Використання АСП у теорії інтелекту приводить до виключення потенційної нескінченності з формальних описів інтелектуальних процесів, що забезпечує краще наближення одержуваних моделей до дійсності, ніж те, яке може дати апарат теорії алгоритмів. АСП вводить у вжиток скінчені алфавітні оператори, адекватніші для формального опису детермінованих, дискретних і скінчених інформаційних процесів, ніж ті що широко використовуються в даний час для тієї ж мети потенційно нескінченні алфавітні оператори.

Важливо підкреслити, що АСП не є чимось зовнішнім стосовно теорії інтелекту. Вона є не тільки засіб формального опису функцій інтелекту, але в її формулах укладена сама суть інтелектуальних процесів. Формули АСП природним чином інтерпретуються як поняття інтелекту, її рівняння — як думки, думки інтелекту, а процес рішення рівнянь — АСП — як діяльність інтелекту.

Наявність АСП відкриває можливість переходу від алгоритмічного опису інформаційних процесів до опису їх у вигляді рівнянь. Ми вважаємо, що мова рівнянь добре відповідає специфіці інтелектуальних процесів. На відміну від алгоритмів і програм для ЕОМ, рівнянням АСП, як, втім, і будь-яким іншим рівнянням, не властива спрямованість дії. Алгоритмічний процес завжди реалізує деяку функцію, з його допомогою проводиться визначення значень цієї функції залежно від значень її аргументів. Рівняннями ж можна задати відносини між змінними. Всі змінні в рівнянні рівноправні, будь-які з них, залежно від нашого бажання, можуть виступати як в ролі незалежних, так і в ролі залежних змінних. Алгоритми і програми описують функціонування систем з входами і виходами, а рівняння АСП описують функціонування багатополісних систем.

В першому випадку вхід і вихід системи жорстко зафіксовані, в другому випадку вхідні сигнали можуть подаватися на будь-кого полюси, а знімання вихідних сигналів також може проводитися з будь-яких полюсів.

Особлива перевага способу опису об'єкту за допомогою рівнянь у порівнянні з алгоритмічним описом полягає в тому, що за рівняннями можна визначити реакції об'єкту при будь-якому способі розподілу вхідних і вихідних сигналів між його полюсами. Якщо число змінних у рівняннях велике (а в рівняннях, що описують інтелектуальні процеси, це число, поза сумнівом, буде дуже великим), то таким же великим буде і число полюсів у об'єкта, описуваного цими рівняннями. Число ж всіх різних способів розподілу вхідних і вихідних сигналів між полюсами об'єкту буде астрономічно великим. У цих умовах практично неможливо створити повний набір алгоритмів, кожний з яких обчислював би реакції об'єкту при якомусь одному способі розподілу вхідних і вихідних сигналів між полюсами об'єкту.

Рівняння вигідно відрізняються від алгоритмів ще й тим, що дозволяють розраховувати (хоча і неточно) реакції описуваного ними об'єкту навіть при неповному його завданні. Не повністю ж розроблений алгоритм буде просто непрацездатним. Опис об'єкта у вигляді системи рівнянь можна легко удосконалювати, проводячи в ньому лише локальні зміни шляхом додавання або виключення окремих рівнянь. Навіть у процесі своєї реконструкції система рівнянь завжди залишається готовою до використання. Описуючи ж реакції об'єкту за допомогою алгоритму, ми ризикуємо зіткнутися з необхідністю корінної переробки алгоритму при щонайменших змінах наших знань про об'єкт. Опис об'єктів за допомогою рівнянь — традиційний спосіб наукового опису світу. Цей спосіб широко поширений у фізиці, хімії, техніці і багатьох інших науках. Нам представляється, що прийшов час використовувати його також і при описі дискретних інформаційних процесів.

Алгоритми і програми для ЕОМ, з одного боку, і рівняння АСП — з іншого, що розглядаються як засоби математичного опису скінчених об'єктів, відрізняються ще в одному вельми важливому відношенні. Алгоритми і програми, описуючи деякий об'єкт, разом із тим містять у собі ще й рецепт обчислення реакцій цього об'єкту на зовнішні дії. Укладач алгоритму чи програми вимушений вирішувати відразу дві задачі: 1) природничонаукову задачу математичного опису об'єкту; 2) формально-логічну задачу визначення реакцій об'єкту на зовнішні дії. Рівняння ж, описуючи об'єкт, не містять у собі рецепту обчислення реакцій цього об'єкту. Щоби ці рівняння привести в дію, їх необхідно розв'язати за допомогою спеціально розробленого методу. При математичному описі об'єкту за допомогою рівнянь їх укладач вирішує не обидві задачі, а тільки одну з них — першу. Він не

робить нічого зайвого, не захаращує опис інформацією, що не відноситься до справи, тому природно чекати, що математичний опис об'єктів за допомогою рівнянь буде простішим, ніж опис тих же об'єктів за допомогою алгоритмів і програм. Ми сподіваємося, що рівняння АСП стануть основним інструментом в справі моделювання інтелектуальних процесів.

Вважаємо, що мова рівнянь добре відповідає специфіці інтелектуальних процесів. На відміну від алгоритмів і програм для цифрових ЕОМ рівняння АСП, як, втім, і будь-яким іншим рівнянням, не властива спрямованість дії. Алгоритмічний процес завжди реалізує деяку функцію, з його допомогою проводиться визначення значень цієї функції в залежності відношень тощо); поняття алгебри логіки і багатозначної логіки; деякі поняття, пов'язані з процесами обробки виразів (слово, довжина слова, з'єднання слів, упаковка виразу, об'єднання виразів тощо); прості лінгвістичні поняття (голосні, шиплячі, елементи парадигматичної таблиці тощо). В цій галузі ще належить виконати величезний обсяг робіт. Їх завершення означатиме формалізацію всієї системи понять, якою володіє людський інтелект. У міру розширення круга понять, математично описаних засобами АСП, її мова ставатиме все більш ємною і абстрактною, поступово наближаючись по своїй силі і виразності до природної мови.

Важливою проблемою теорії інтелекту є розробка методів і засобів для вирішення рівнянь АСП. Вони необхідні для фізичного відтворення інформаційних процесів, формально описаних теорією інтелекту. Без вказаних методів і засобів математичні моделі функцій інтелекту залишаться тільки мертвими записами на папері. На даний час зроблені перші кроки у напрямі рішення цієї проблеми: побудовані алгоритми рішення рівнянь АСП шляхом приведення їх до досконалої диз'юнктивної нормальної форми і шляхом послідовної підстановки в рівняння знайдених значень змінних; розроблені методи мінімізації формул і методи декомпозиції рівнянь АСП; знайдені способи явного виразу невідомих для рівнянь із постійними і змінними параметрами; описаний ряд тотожностей, на основі якої можуть виконуватися спрощення формул за наявності в них символів вищого рівня абстракції, що скорочують (заперечення, імплікація, об'єднання множин, з'єднання слів, упаковка виразів тощо).

Рішення рівнянь, що описують складні інтелектуальні процеси, можна подати як багатоярусний процес, здійснюваний на різних рівнях абстракції. Щоб повністю автоматизувати процес, необхідно заздалегідь виконати дуже велику роботу щодо формального опису властивостей понять різного рівня абстракції, створити теорію тотожних перетворень і спрощення формул різних консервативних розширень АСП. Як перший крок в цій галузі розроблені початки теорії виразів, в якій формально описані

прості перетворення послідовностей символів (селекція, комутація, зсув, перестановка, вибірка, розмноження, знищення, заміна букв виразу, підрахунок букв виразу, віднімання виразів тощо).

Насущною задачею теорії інтелекту є створення універсального вирішувача задач, який на першому етапі робіт можна подати у вигляді спеціальної програми для цифрових ЕОМ, розв'язок певного класу рівнянь АСП з символікою достатньо високого рівня абстракції. Рішення цієї задачі дозволило б взагалі відмовитися від складання програм для багатьох задач, складність яких не перевершує деякий заданий рівень. Достатньо б було записати умови задачі у вигляді системи рівнянь АСП і вирішити систему за допомогою універсального вирішувача.

Другий етап робіт у цьому напрямі представляється як створення спеціалізованої машини для вирішення рівнянь АСП заданого рівня складності. Така машина, по суті, виявилася б відправним пунктом в справі створення системи штучного інтелекту. У міру того як у процесі вдосконалення машини збільшувалася б гранична складність вирішуваних нею рівнянь, а також росла б рівень абстрактності і число освоєних нею понять, спостерігався б поступовий розвиток її інтелекту. Зараз зроблені певні зусилля у напрямі створення передумов до побудови такої машини. Розроблені початки теорії ланцюгів перемикачів і теорії цілком скінчених автоматів, розглянуті способи схемної реалізації деяких первинних математичних структур і операцій (множин і відношень, операцій над ними, булевих і перемикачів функцій, арифметичних дій, операцій над виразами і словами тощо).

4. Морфологічний аналіз

Морфологічний аналіз — це оброблення слів форм без зв'язку з контекстом. Словоформа — це відтінок тексту між двома просвітками. Розділові знаки вважаються окремими словоформами.

У центрі уваги морфології стоїть слово з його граматичними змінами. У лінгвістичній літературі термін “морфологія” вживається для позначення відповідної системи всередині мови, у яку входить словозміна (словотворення розглядається як самостійна система) [1].

Під словозміною розуміється утворення форм одного й того ж слова, що існує в мові, як система форм (словоформ). Словоформи «можна визначити, як регулярні видозміни слова, об'єднані тотожністю його лексичного значення й неоднаковими морфологічними значеннями» [1]. Запропоноване визначення охоплює такі зміни, за яких зберігається цілим лексичне значення слова.

Нижче об'єктом формального опису є письмова, тобто орфографічна, а не фонематична реалізація словоформ української мови. Така форма запису усуває необхідність додаткового оброблення під час введення/виведення текстів, а також дозволяє усунути із опису

акцентний компонент. При цьому, окремою словоформою є відтінок тексту між двома проміжками.

Утворення слова звичайно відбувається за допомогою спеціальних словозмінних морфем. Під морфемою будемо розуміти мінімальну значущу одиницю мови: «Морфема — це найменша неподільна значуща частина слова є одиниця мови, а в мові виступають окремі представники морфем, які зручно називати морфами» [1].

За значеннями і функцією в структурі слова морфемі поділяються на кореневі і службові. Морф — конкретний вияв морфемі, найменша значуща частина, що виділяється у складі певної конкретної словоформи слова. Особливу роль під час словозміни відіграють закінчення слів форм (флексії), що передають, як правило, граматичні значення в українських словах. Закінчення завжди виступає в певній системі словозміни змінюваних частин мови.

Типи відмінювання іменників, прикметників, числівників, займенників розрізняються системою закінчень, які виражають граматичні значення відмінка, роду і числа або лише відмінка (в кількісних числівниках). Наприклад, у іменників флексії завершують матеріальну побудову, закріплюють лексико-граматичне значення слів, сприяють відображенню таких категорійних ознак як рід, одухотвореність, число, відмінок. Дієслова мають складну систему дієвідмінювання за особами, родами і числами, основними показниками яких є особово-, родово-числові флексії.

Будемо класифікувати флексії на регулярні та нерегулярні. Регулярні флексії — найрозповсюдженіші й приймаються за норму. Описана нижче математична модель базується на формальній моделі морфології української мови.

У морфології української мови існує певна залежність (відношення) між флексією та оточуючим її текстом. Завдання полягає в математичному описі існуючої залежності, тобто у формалізації поняття флексії. Текст, що охоплює закінчення, є неоднорідним щодо нього. Будемо розрізняти найближчий текст (що межує безпосередньо із закінченням у словоформі) і подальший текст, (що межує з словоформою, у якій перебуває закінчення, що нас цікавить).

У роботі [1] введено поняття текстового відношення $T(X, Y)$, де X — набір граматичних ознак, що виділяє підмножину усіх текстів із деякої множини різноманітних рядків обмеженої довжини, упорядкованих із букв скінченого алфавіту; Y — текст. Текст розглядається як багаторівнева структура, у якій із букв утворюються морфи, а з морфів — словоформи, із словоформ — речення, та з речень — абзаци. Окремі частини цієї конструкції названі фрагментами тексту (букви, морфи, словоформи, речення тощо). У математичній моделі системи української словозміни, як фрагмент тексту виступають словозмінні морфемі, що можуть бути суфіксами й закінченнями. Враховуючи висловлене

раніше зауваження щодо ролі флексій у процесах словозміни, зосередимо подальший розгляд на методиці формального опису процесів зміни слів (за допомогою флексій), що можна застосувати й до інших лінгвістичних одиниць тексту.

Дію тексту, що оточує закінчення (як ближчого, так і подальшого), замінимо уведенням деякого набору ознак, який проінтерпретуємо як *смысл закінчення* [1]. Під смыслом будемо розуміти набір проміжних змінних, що зв'язують рівняння, які описують закінчення, з рівняннями, що описують решту частини тексту. Набір значень цих змінних будемо інтерпретувати як значення смыслу. Смысл розглядається як змінний вектор $X = (X_1, X_2, \dots, X_m)$, де m — число компонентів вектора смыслу, а змінні X_1, X_2, \dots, X_m — компоненти смыслу.

Нехай Y_1, Y_2, \dots, Y_n — закінчення, — де значенням Y_i служить i -а за рахунком буква закінчення. Набір $Y = (Y_1, Y_2, \dots, Y_n)$ буквених змінних Y_1, Y_2, \dots, Y_n назовемо *змінним закінченням*, набір значень цих змінних — *індивідуальним закінченням* чи просто закінченням. Змінні змінного закінчення, їх значення назовемо значеннями компонентів закінчення.

Під час побудови математичної моделі флексії описується *фрагментне морфологічне* відношення $L(X, Y)$, що відтворює зв'язок між смыслом X і змінним фрагментом (флексійним морфом) тексту Y . Параметрами уведеного відношення можуть бути будь-які із компонентів вектора смыслу. При цьому $X \in H, Y \in M$, де H — множина наборів значень компонентів смыслу, M — множина різних послідовностей скінченої довжини, упорядкованих із букв українського алфавіту та символу проміжка (L —), що служить для заповнення вільних від букв позицій у закінченнях.

Вважаємо, що морфологічне відношення $L(X, Y)$ виконується тоді і тільки тоді, коли Y передає значення компонентів смыслу X . Мовна практика говорить про те, що результат перевірки умови виконання морфологічного відношення повинен бути однозначним (у протилежному випадку людина не змогла б ефективно формувати тексти).

Згідно з викладеним вище принципом однозначності, закінчення завжди однозначно залежить від смыслу. Цей принцип можна інтерпретувати як вимогу повноти набору ознак (компонентів смыслу), із допомогою якого здійснюється вибір індивідуального закінчення. Набір ознак будемо називати повним, якщо він при будь-яких значеннях ознак забезпечує однозначність вибору відповідної йому флексії. У якості смыслу приймемо будь-який набір ознак, що задовольняє вимозі повноти. Під повним смыслом флексії будемо розуміти набір, складений з деяких компонентів вектора смыслу X . Смысл флексії, у прийнятому раніше значенні, будемо називати *повним смыслом*. Поняття повного й неповного смыслу флексії уведені за аналогією з відповідними поняттями

для фрагменту тексту. Моделювання фрагментного відношення необхідно почати з вибору структури багатоконпонентних векторів X та Y . Як компоненти вектора Y необхідно вибрати букви закінчення. При цьому нумерація букв ведеться в межах флексії зліва направо. Що ж стосується вибору структури вектора смыслу X , то тут значно все складніше.

Для того, щоб отримати компактний та повний опис фрагментного морфологічного відношення $L(X, Y)$, необхідно вибрати компоненти вектора X із урахуванням внутрішньої структури мови. Для раціонального вирішення цього завдання будемо спиратися на поняття, сформульовані морфологією української мови. Критерієм під час оцінки припустимості вжитку того чи іншого поняття, виробленого мовознавством, повинен бути, очевидно, успіх формалізації природної мови. Через те, що об'єкт, який моделюється, є достатньо складною для математичного опису системою, зручно розбити його на ряд підсистем (які будемо надалі називати «мовними блоками»), що пізніше об'єднуються теоремою про розкладання в алгебрі скінчених предикатів [1].

Для кожного мовного блоку фіксують значення деяких із компонентів смыслу, які ми назовемо обмеженими. Наприклад, одним із таких компонентів смыслу може бути ознака частини мови, що дозволяє формалізувати флексійне оброблення кожної частини мови окремо. Другу групу ознак, що служать безпосередньо для вибору закінчення усередині деякого мовного блоку, назовемо змінними. Необхідно відзначити, що назви двох груп ознак умовні, оскільки насправді всі компоненти вектора смыслу є змінними. Обмежувальним ознакам поставимо у відповідність змінну ξ , а змінним — X . Таким чином, маємо можливість уточнити структуру вектора смыслу

$$X = (\xi, X) = (\xi_1, \xi_2, \dots, \xi_k, X_1, X_2, \dots, X_l),$$

де k, l — число обмежувальних та змінних компонентів вектора смыслу відповідно.

Змінні ознаки в свою чергу неоднорідні щодо закінчення. Раніше вже підкреслювалась неоднорідність тексту, що оточує флексію. Впливові ближнього тексту на закінчення будуть відповідати лексичні ознаки, а подальшого — синтаксичні.

Щоби визначити склад змінних ознак математичної моделі, необхідно сформулювати таку вимогу до неї: закінчення Y , яке перебуває в морфологічному відношенні зі смыслом X , повинно однозначно визначатися смыслом (у відповідності з принципом однозначності). Унаслідок цього, до складу змінних компонентів вектора смыслу включенні як словозмінні категорії (для прикметників — рід, число, відмінок), так і несловозмінні, але такі, що керують утворенням різних варіантів форм. Наприклад, наявність у деяких слів сучасних та архаїчних варіантів форм уста, перст — *архаїчне*, губи, палець — *сучасне*. Тому до складу змінних ознак включена ознака зі значеннями: сучасне (с), архаїчне (а). Не завжди

закономірності в утворенні варіантів форм настільки очевидні, що їм можна поставити у відповідність будь-яку морфологічну категорію. У подібних випадках будемо уводити так звані умовні морфологічні категорії, що відповідають простому перерахуванню варіантів форм.

Отже, завдання полягає у формальному описі засобами АСП фрагментного морфологічного відношення $L = (\xi, X) = (\xi_1, \xi_2, \dots, \xi_k, X_1, X_2, \dots, X_l, Y_1, Y_2, \dots, Y_n)$, що буде моделлю флексійного оброблення українських слівформ. Морфологічне відношення визначається своєю характеристичною функцією $L_\varphi(X, Y)$, що задається наступним чином:

$$L_\varphi(X, Y) = \begin{cases} 1, & \text{якщо } X \text{ і } Y \text{ зв'язані морфологічним відношенням;} \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Відношення (6) є предикатом, визначеним на скінченій множині предметної галузі. Він дозволяє інтерпретувати різноманіття граматичного оброблення українських флексій (аналіз, синтез, нормалізація, корегування помилок тощо), із допомогою розв'язків канонічних рівнянь виду $L_\varphi(X, Y) = 1$.

Подамо морфологічну функцію у явному вигляді:

$$y_j^\sigma = F_\sigma(\xi, X), 1 \leq j \leq n.$$

Вона буде описана, якщо нам вдасться подати у вигляді формул предикати F_σ , де σ належить множині значень, які приймають букви закінчень. Предикати $F_\sigma(\xi, X)$ задаються різноманітними ідентифікаторами (розпізнавачами) y_j^σ для змінної y_j .

Тут прийнято паралельний метод опису множини флексійних морфів [1], згідно із яким утворюється множина місць для усіх флексій, що підлягають опису. Кожному місцю j ставиться у відповідність своя змінна y_j . Як формальний еквівалент флексії приймаємо набір значень усіх змінних y_j . Значенням змінної y_j є стан місця j для індивідуального морфа.

Таким чином, задача опису множини флексійних морфів зводиться до формального опису функції виду

$$y_j^\sigma = F_\sigma(\xi_1, \xi_2, \dots, \xi_k, X_1, X_2, \dots, X_l), 1 \leq j \leq n.$$

Висновки

1. Сфери предметних галузей, де найдоцільніше працювати з даними і знаннями, поданими мовними моделями — це галузі з переважанням емпіричного знань, де складність фактів і їх описів виключає використання мови математики і зокрема інформаційно-інтелектуальних технологій та інтелектуальної підтримки суспільно-економічних процесів діяльності фінансово-кредитних установ.

2. Найбільш загальна проблема побудови системи управління семантичного чи семантико-прагматичного рівня взаємодії, пов'язана з вибором технології контекстно-залежного подання знань, побудовою інформаційних баз (даних і знань) про предметну галузь і механізму висновку для отримання необхідних рішень.

3. Єдино відомим нам, об'єктивним носієм знання та інтелекту є людина, а виразником, засобом до зовнішнього спілкування та носієм інтелекту є людська мова, що й складатиме об'єкт і напрямок досліджень у наступних частинах роботи.

4. Математична структура даних у декларативних мовах базується на системах предикатних рівнянь у алгебрі скінчених предикатів або на аксіоматичній теорії множин, у якій теорія множин інтерпретується як структура даних.

5. Проблема створення ШІ полягає не в тому, щоб «будувати штучних людей», а в тому, щоб пізнати природні організми настільки, щоб використовувати їх на рівні систем, тобто в керованому або коректованому режимі, зробити їх керованими.

6. Кожний атрибут системи ШІ сам може бути універсальним функціональним перетворювачем, але іншого ієрархічного рівня. З цього випливає, що універсальний функціональний перетворювач є системою рекурсивною.

7. На підставі вивчення даних фонетики і орфографії української мови розроблено:

- методу формального опису букво-фонемних відношень, побудовані математичні відношень голосних та приголосних звуків української мови, на основі яких формалізуються закономірності поєднання голосних і приголосних букв та звуків, що утворюють пари за дзвінкістю-глухістю й твердістю-м'якістю, а також поза парних приголосних;

- математичну модель акцентних характеристик української мови. Усі ці результати дозволяють отримати компактну модель для економного запису словників природної мови в пам'яті ЕОМ.

8. Із допомогою опису букво-фонемних зв'язків у вигляді системи рівнянь АСП можна:

- в інтелектуальному інтерфейсі здійснити формальний перехід від фонетичного запису слова до його графічного зображення (задача фонетичного аналізу);

- за даним орфографічним текстом отримати його фонетичну транскрипцію (задача синтезу мови);

- перейти до системного та структурного синтезу процесорів функціонально-зорієнтованих на аналіз та синтез мови, які є необхідною складовою частиною систем штучного інтелекту.

Список літератури:

- [1] Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г. Основи теорії синтезу надшвидкодійних структур мовних систем штучного інтелекту. — К.: ІЗМН, 1997. — 264 с.
- [2] Бондаренко М.Ф., Коноплянко З.Д., Четвериков Г.Г. Основи теорії багатозначних структур і кодування в системах штучного інтелекту. — Х.: Фактор-Друк, 2003. — 336 с.
- [3] Шабанов-Кушнарченко Ю.П. Теория интеллекта. Математические средства. Технические средства. — Харьков: Вища шк., 1984, 1986. — 144 с., 136 с.

Надійшла в редакцію 23.06.2022



Ilona Revenchuk¹, Yevhen Ostashko²

¹ Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
ilona.revenchuk@nure.ua, ORCID: 0000-0002-5188-9538

² Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
yevhen.ostashko@nure.ua

MINIMIZING THE COSTS OF BUILDING COMPLEX CROSS-PLATFORM APPLICATIONS WITH FLUTTER AND FIREBASE

In our time one of the most if not the most common types of services is client-server ones. There are many frameworks written in many programming languages for both client applications and server ones, especially considering many different popular operating systems that are used by people today. With this many options, the question of which technologies to use to achieve the goal with minimum sacrifices is always relevant and in this article, we research what might be one of the most convenient technologies for client-server development not only for applications with relatively small amount of users but huge ones as well that allows for 1-2 developers to relatively quickly develop, release and maintain big systems with multiple client applications.

As a result of the work, there has been performed a research on the technology for developing cross-platform applications – Flutter, the easiness of its use, and its advantages compared to alternatives, on Firebase services for minimization of the costs to customize the server and on the combination of these technologies for modern client-server application development.

INFORMATION SYSTEM, SERVERLESS, FIREBASE, FLUTTER, DART, CLEAN ARCHITECTURE.

І.А. Ревенчук, Е.В. Осташко. Мінімізація витрат на створення складних кросплатформових додатків за допомогою Flutter та Firebase. У наш час одним із найпоширеніших, якщо не найпоширенішим видом сервісів є клієнт-серверні. Існує багато фреймворків, написаних багатьма мовами програмування як для клієнтських програм, так і для серверних, особливо враховуючи багато різних популярних операційних систем, якими користуються на сьогоднішній день. З такою кількістю варіантів питання про те, які технології використовувати для досягнення мети з мінімальними витратами, завжди актуальне і в цій статті ми досліджуємо технології, які можуть бути одними із найзручнішими для клієнт-серверної розробки не лише для додатків з відносно невеликою кількістю користувачів, але і для великих, що дозволяють 1-2 розробникам відносно швидко розробляти, випускати та підтримувати великі системи з кількома клієнтськими програмами.

За результатами роботи було проведено дослідження технології розробки кросплатформних додатків – Flutter, простоти її використання та переваг перед альтернативами, сервісів Firebase для мінімізації витрат на налаштування серверної частини та взаємодії цих технологій для розробки сучасних клієнт-серверних програм.

ІНФОРМАЦІЙНА СИСТЕМА, БЕЗСЕРВЕРНА, FIREBASE, FLUTTER, DART, ЧИСТА АРХІТЕКТУРА.

Introduction

Cross-platform development is the most effective when it comes to client applications since you use one code base for multiple applications simultaneously increasing amount of people the app is available for and amount of people that are needed to develop it. Fewer code results in fewer bugs, their faster resolution, and quicker implementation of new features. Respectively fewer hours of work are required for the whole cycle of development that results in the same goal being achieved with a lot fewer resources than if each application had been developed by separate teams using native development frameworks. This approach has been spreading rapidly since its first introduction and is one of the most popular ones at the time of the article.

Today many such cross-platform frameworks exist and so the problem of choosing the most suitable one arises again. Different frameworks support different operating systems, use different programming languages, and have communities of different sizes. That is where the Flutter framework, that we are researching in this article, shines in theory, not only is it the most modern cross-platform

framework at the time of the article, which means it incorporates the most modern approaches in software development it also supports 6 of the most popular operating systems with the single codebase (Android, iOS, Web, Windows, Linux, MacOS) [1]. The practical side of this framework is what needs to be researched to find out if the framework can be used in most use-case scenarios.

For the client-server application, the client alone is obviously not enough so the question arises of what technology to use for the server side. Just like with a client application, many options are available for back-end development, but the simplest one of all is to use the so-called “serverless” approach. That approach means using existing services such as Amazon web services or, in our case, Firebase, that do most of the work, such as scaling, security, and open API for us. Not having to set up and program the server site has its disadvantages, which we will talk about later, but in most cases, such an approach saves a lot of time and money speeding up the development process and moving release deadlines earlier. Since configuring the server using the “serverless” approach does not require coding, minimum training and education are

required so the same experienced developers that are developing the client application can take on this job as well subsequently making the need for front-end and back-end teams to use the time to communicate and sync their work non-existent.

1. How Flutter works

During Flutter development, applications run on a virtual machine, which allows you to reflect changes in the codebase without the need to completely recompile the application, which speeds up the development significantly, especially compared to native mobile development (Android/iOS). For the release version, Flutter applications are compiled directly into machine code, regardless of whether it is on Intel x64 or ARM, or even on JavaScript if we compile for the web, and as a result, we get an installation file specific for the target platform. The framework is open access and has an active ecosystem of third-party packages that add functionality to the basic framework [2][3].

1.1. Structure

Flutter exists as a series of libraries divided into different layers (pic. 1) that can be replaced if needed [4]. No layer has privileged access to the layers below it, and every part of the framework is designed to be replaceable or completely removed as needed.

The lowest layer, called Embedder, is the layer that procures the cross-platform capabilities of the Flutter framework. It's written using the programming language corresponding to the target platform (Java and C++ for Android, Objective-C for iOS, etc.), so each platform that Flutter supports has its own embedder. This layer ensures the native thread setup for Flutter to work and contains native plugins to allow the framework to use the platform's native capabilities, such as a camera for mobile and many more. Also, this layer converts the application interface configuration, provided by the higher layers, into native drawn elements.

The next layer above the Embedder is a Flutter Engine that is written in C++ and supports primitives, required for the Flutter applications. The engine rasterizes the composite scenes each time a new frame needs to be rendered. It provides a low-level implementation of Flutter's core interface, including graphics, text composition, file system access, plugin architecture, and a Dart compilation toolchain. The engine is available for frameworks through the corresponding package – dart:ui, which wraps the underlying C++ code in Dart classes.

The top-level layer is the one developers interact with the most – the Framework layer. This layer contains various libraries for interface build that are abstractions to the lower-level components to ease the development process, speeding it up consequentially. If going from the lower level of the Framework to the higher the following structure is present:

- fundamental classes and building blocks such as animations, drawing, and gestures;
- rendering layer, which provides abstractions for working with the layout;
- widget layer, which is an abstraction of the composition. Each rendering object from the rendering layer has a corresponding widget class, and this layer also allows you to define combinations of widgets that can then be reused;

Material and Cupertino libraries that provide widgets typical of Material and iOS design languages.

1.2. Rendering

When analyzing cross-platform frameworks, not the last thing they pay attention to is its performance. Surprisingly, Flutter's rendering technology gives it a level of performance comparable to a single-platform framework. In the example of mobile cross-platform frameworks, an abstraction is usually created over the interface libraries of specific platforms. These native libraries usually convert classes to a view on a Canvas object. The interaction of cross-platform framework code with native creates many additional processes that slow down the application. Flutter, in turn, minimizes such abstractions by not using native rendering libraries in favor of its widget structure. The Dart code that draws the Flutter app view is compiled into native code that directly uses drawing on the canvas. Flutter uses Skia technology for rendering, just like Android. The framework engine also includes its own private Skia repository, allowing developers to update their applications with the latest improvements without having to update the native operating system version.

The principle of Flutter is “what simple is fast” [5]. The flow of data from the event to the rendering instructions can be seen in fig. 2.

So, the rendering process goes like this:

- when Flutter wants to redraw its fragment in the corresponding widgets, a special “build” method is called;

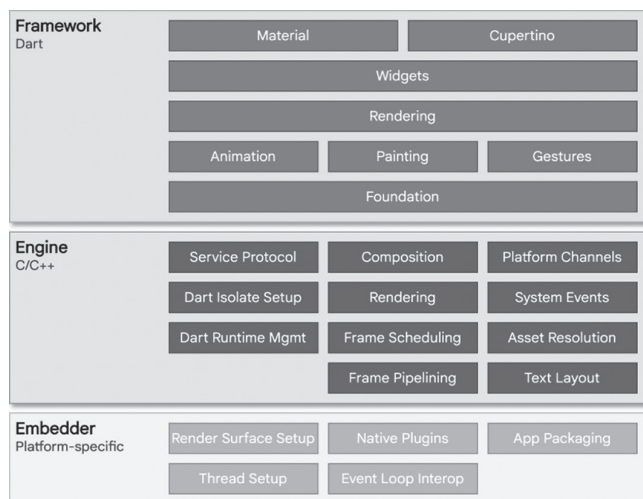


Fig. 1. Flutter's architectural layers

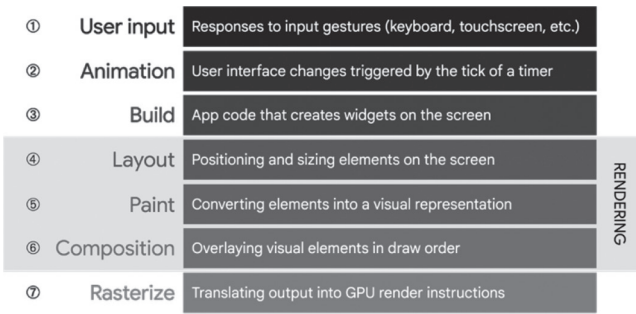


Fig. 2. Rendering pipeline

– during the build phase, Flutter converts the widget structure into a corresponding tree of elements (fig. 3) with one element per widget. Elements are divided into component elements, which simply contain other elements, and elements of render objects, which participate in the drawing phase;

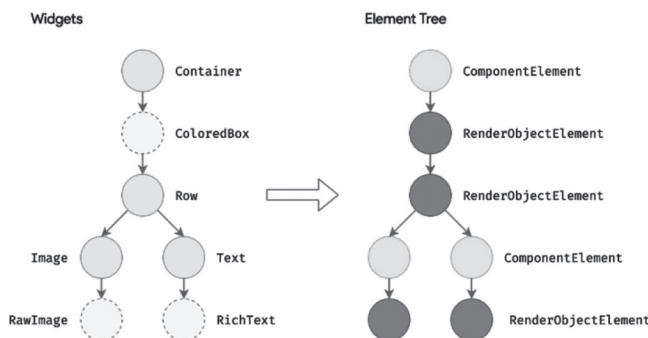


Fig. 3. Transformation of the widget's structure into a tree of elements

– during the construction phase, Flutter creates or updates an object that inherits from RenderObject for each element of the render object in the tree (fig. 4);

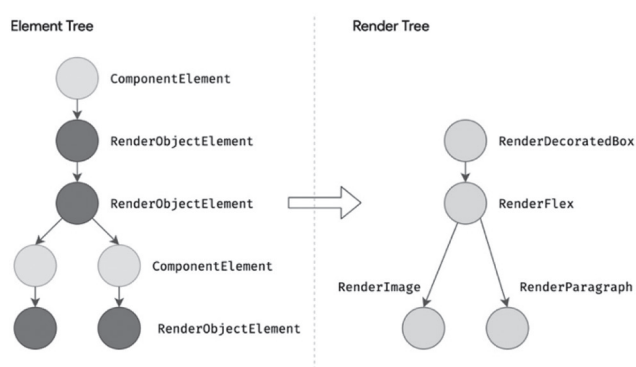


Fig. 4. Transformation of the elements' tree into a tree of render objects

Next, the framework traverses the render tree depth-first and passes down size constraints from parent to child. Then the descendants respond by passing their sizes to the parents, which must be within the received limits (fig. 5). After that, each object is ready for drawing.

The root of all render objects is the RenderView, which represents the full output of the render tree.

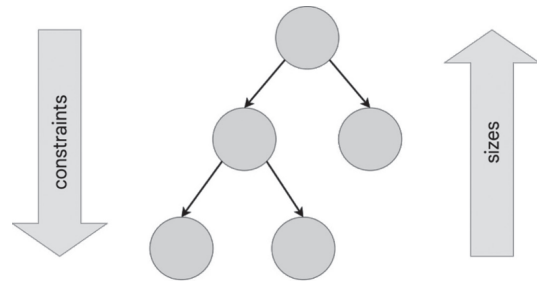


Fig. 5. Passing constraints and dimensions in the render object tree

2. Firebase services

Flutter contains many special services such as Machine Learning, that are not typically used in an average service so they will be omitted and the most useful and common ones will be described. The Firebase architecture consists of several components that interact with each other to provide a framework for developing and hosting web and mobile applications [6]. The following are useful for our purpose:

– Firebase Cloud Storage is an object storage that is available on Google Cloud platforms. When this service is added to Firebase, the application gets access to Google's security measures and the ability to safely download or upload data;

– Cloud Firestore is a scalable, flexible database service for server, mobile, and web development. It serves as a NoSQL document database. Provides "out of the box" access to web sockets, that is, the ability to reactively update data in the client application on database changes. It also has its own system of security rules, which is written in a JavaScript-like programming language and controls access to collections and documents according to user needs;

– Authentication – can be used for simple user authentication through login and password, mail, phone number, and various social networks;

– Hosting – hosting service for web content. It can be combined with cloud functions to develop and host your own microservices;

– App Distribution – allows you to distribute pre-release versions of applications and configure a list of testers for them;

– Cloud Functions – provides an opportunity to create both HTTP endpoints and trigger functions that respond to prescribed events, such as creating a document in a certain collection. Useful when services that are needed to implement some functionality in the application have webhooks, for example, the Stripe payment management service;

– Remote Config is a key-value type cloud data storage for managing some flags and parameters necessary for the client application, for example, the minimum version of the application to which the user will be forced to update the application;

- In-App Messaging allows you to create a system for sending push messages to device tokens, usually stored in Cloud Firestore in the user’s document. It is especially useful on the mobile platform, where push notifications are most common;

- Crashlytics – allows you to track errors in applications connected to the service. May contain error codes and messages, and also tracks app crashes;

- Performance – allows you to see how expensive certain processes in the application are, on which platforms, and with which data. Useful to use for complex functionality to test the impact of changes on performance;

- Google Analytics – allows you to create many events that can happen in the application and analyze them with the help of real users. In this way, it is possible to highlight the priority functionality, which is used by almost no one, and plan the development based on real data;

- Dynamic Links – allows you to develop so-called “deep” links that will send the user to, for example, a corresponding mobile application, where it will be processed, instead of a web page;

- AdMob allows you to add integrated Google advertising to the application, immediately providing a method of monetizing the service.

All of the Firebase services listed above would be useful for a typical service providing not only the means of implementation of the basic functionality but supporting services to help improve the user experience as well.

3. Flutter-Firebase communication

Since both Flutter and Firebase belong to the same company – Google, the interaction between these technologies is made as accessible as possible [7]. It is also important that any updates and changes in Firebase will be adapted to the Flutter context in priority, which accelerates the improvement of the service thanks to new functionality and bug fixes.

Flutter has all the necessary libraries for a comfortable interaction of the client application with Firebase services on the website, which distributes both custom and official packages for the framework (pub.dev). They abstract the work with the Firebase API, which makes it much easier to use these services without creating a mechanism for working with Rest, as you would have to do with a regular server. The same packages can also be found for most modern frameworks such as React, View.js, Java Spring, etc.

4. Flutter’s performance

Before jumping into using the Flutter framework with all of its stated incredible capabilities a company should still research all of the problems the framework may produce and the most common question is if the productivity tradeoff is small enough to warrant the use of the framework instead of the native solutions, that has been present on the market for decades and has been tested and

improved with time more than any other solution for the platforms.

The Flutter framework makes the whole development process many times easier, faster, and consequentially – a lot cheaper, but if the result product’s productivity is interfering with users’ experience it can make this product very unsuccessful on the market, making all of the spent resources on the development go to waste and the saving of a big part of them, choosing a cheaper development process, a bad decision that led to nothing but losses. That risk is the reason why relatively new technologies usually take years to be incorporated into big companies and are mostly used by small companies that do not have a lot to lose. In a sense, these small companies are stress-testing these technologies on real projects for big companies to incorporate if proven effective.

Flutter is not an exception, though its rapid growth in popularity is superior to most, not least because its origin is one of the biggest companies in the world – Google, which also used it to rewrite some of their popular, extensively used products, such as Google Pay. These products are performing incredibly in production and have been proven to be as effective as their native counterparts with the benefit of a lot simpler codebase, which inclined many to try Flutter for their projects.

Real big projects written using Flutter prove its effectiveness by themselves, but knowing exactly how different Flutter’s productivity is compared to the native apps is essential, especially if the project has to be as fast as possible, for example, a video game would need every frame per second it can get for the most optimal experience of the user even on the old devices.

The precise comparison of the Flutter and native Android apps has been done and published in an article before [8] and it presents a relatively accurate representation of the productivity differences between the Flutter and other platforms’ native SDKs since the technology, that Flutter uses, is practically the same for every platform and boils down to drawing elements on the platform provided canvas.

In the mentioned article the study concludes with the native app performing faster than a Flutter one in every aspect, except for the threads rendering, the fact that there is an aspect of the cross-platform framework that works faster than the native counterpart is amazing, but the rest of tests resulted in the Flutter’s loss. But that does not mean that Flutter’s productivity is an issue, since looking closely into the collected by the author data the only big difference in performance, that can be spotted by the naked eye is app startup, where Flutter seems to be performing heavy tasks, the rest of tests show a minuscule difference of average 50ms difference and the most common tasks that users will face perform with the difference in a smaller range of 10-30ms.

In conclusion, the Flutter apps perform efficiently enough for the absolute majority of cases in the app development sphere, since most apps tend to not perform complex user interface tasks.

It is also important to note that the huge Flutter popularity led to a rapid increase in the corresponding team's funding and the fact that it's an open-source project allowed countless developers to improve different aspects of the framework themselves. Many of those changes have been officially incorporated into the framework's stable version after the review, so Flutter's efficiency increases very fast compared to most other frameworks which means the productivity may increase further in the future.

5. Cloud Firestore

Since Cloud Firestore has a role of a database, which would be storing most of the data of the absolute majority of apps using it's important to know how it works and how productive it is compared to the alternatives [9][10].

5.1. Servers' locations

When it comes to the productivity of a server the first thing that comes to mind is its location. The closer the server is to the user the faster the response time will be. When choosing the location for the Google Cloud Platform services, which include Cloud Firestore, the developer is presented with two options: a multi-region location or a regional location.

Multi-regional location maximizes the availability and durability of your database. It stores data replicas on multiple servers across multiple regions, some regions, called witnesses only participate in the process of replication. This method allows having consistent data and request times across multiple regions, which is recommended for global apps. At the time of the article, 2 multi-regions are available: Europe and the United States.

Single region location maximizes the efficiency and cost of the data transfer for the specific region. Using a single region location would be more effective for the apps, that are not designed to be global and are used in a specific region, since request time increases the farther you are stationed from the selected region. Multiple regions are available across America, Europe, Asia, and Australia.

Since Firebase charges the user by the number of documents you read, write and delete obviously a multi-regional location would charge more for every operation since the data needs to be replicated to support effective data accessibility across multiple regions.

5.2. NoSQL Advantages and Limitations

Since Cloud Firestore is a NoSQL document-oriented database it is important to understand the principles behind such a data-storing approach since it would not be effective for every service to use it.

The main characteristics of NoSQL Databases architectures are [11]:

- schema-less structure;
- permitting data representations to grow effectively and dynamically;

- horizontal scaling, by data replication of the collections and sharding over massive clusters.

The main advantages of NoSQL Databases:

- volume: data at rest – terabytes to exabytes of existing data to process;

- velocity: data in motion – streaming data, milliseconds to seconds to respond;

- variability: data in many forms – structured, unstructured, text, etc.;

- veracity: data in doubt – uncertainty due to latency, deception, ambiguities, etc.;

- not built on tables and does not employ SQL to manipulate data;

- can handle unstructured, messy, and unpredictable data;

- helpful for working with large sets of distributed data.

The disadvantages are the following:

- disorganized data, it's harder to query needed data;
- the lack of JOINS – the lack of relations between data makes the developer send multiple requests to access some data;

So, in conclusion, if the data for the service needs a strict structure for maximum efficiency it would be recommended to use a conventional relational SQL database, but that does not mean that Cloud Firestore cannot be used for such cases. Even though the documents' structures are not strict and inside one collection documents with different structures are allowed to occur the structure can be enforced manually through the developers' efforts.

In the Cloud Firestore on the top level are only collections, each collection can contain only documents, but each document can contain a subcollection. The structure can be achieved through the client application's effort by defining strict models and paths for data to be stored. That does not solve the problem of fetching multiple different documents through one request but it makes it possible to predict the documents' contents and locations. For most tasks that may occur in a typical service that should be enough.

6. Experiment

Any theory needs to be tested in practice to have value as research. In our case, a test service with a client application written using the Flutter framework that uses most Firebase services to function needs to be developed and tested.

6.1. Experiment steps

The following tasks need to be performed:

1. basic Flutter application setup that includes app architecture, presentation layer architecture for state management, navigation, and dependency injection setups;

2. check how native applications for supported platforms look and feel;
3. create a Firebase project and connect it to the Flutter app;
4. add registration and authentication using the Firebase authentication service since it's the most common use-case;
5. add database usage with Cloud Firestore, Cloud Storage and Real-time database, test their security, add rules;
6. add Cloud functions: HTTP ones and triggers for collections' documents' states changes;
7. add Remote Config usage, push notifications using In-App Messaging, dynamic links and adds using AdMob;
8. add analytics using Google Analytics, Crashlytics, and Performance;
9. distribute mobile apps using app distribution and host web apps using Hosting.

Every listed development task needs to be tested on the speed of implementation, its complexity, and if the result is satisfactory in terms of performance and security.

6.2. Experiment results

As a result of the experiment, a Flutter test application has been developed with Firebase services covering the need for a database, file storage, authentication, etc. Below the experience from the development process of each experiment step in the order presented in the previous section is listed.

6.2.1. Flutter application architecture

The Flutter community has created many libraries that make initial project setup and things like navigation, dependency injection, and interface state management easier. For the test project, a `flutter_modular` package has been chosen as it makes implementing a modern app architecture easier. The package allows us to divide our app into different, separate modules, each module has its own injected dependencies and they are injected and disposed of depending on the navigation state, so if there are no screens, that belong to a specific module, in the screens stack the dependencies, if were injected prior, are disposed of.

During development, there were discovered 2 things that `flutter_modular` lacks, that needed to be implemented separately by downloading package and changing it locally.

First, nested navigation, which allows us to open screens inside other screens, that we use for the main screen with the bottom navigation bar does not have caching. That means each tab is loaded every time the user navigates to it, which is a bad practice as many unnecessary requests are sent.

Second, there was no ability to push several screens at the same time. This is crucial when it comes to opening dynamic links and push notifications, as when we lead the

user to the corresponding screen the screens back-stack needs to be filled with the correct items.

The basic app structure in a final solution is as follows: an app is divided into features, and each feature has its own package, so it's physically separated from the other features. The feature has 3 packages inside for each layer according to the clean architecture principles. The presentation layer also has a navigation package, that contains the required info for other features to navigate into it if needed.

For the presentation layer architecture for state management, a BLoC (business logic component) was chosen since it's the most popular solution at the moment and perfectly solves the problem of separating the UI components and business logic.

The version of Dart at the time of the article does not have so-called sealed classes like, for example, Kotlin programming language does. But surprisingly it could be solved by the code-generation package called `freezed`. It automatically generates boilerplate code for models, such as serialization/deserialization, and comparison, and allows the addition of a similar to sealed classes behavior to a class. This has been extensively used for data transfer objects and navigation information so that we can pass needed parameters to a screen through such a class. Other code generation packages were also used, for example, to implement localization only JSON files with strings needed to be filled manually, and the needed code was generated with a corresponding command.

6.2.2. Result client applications

The Flutter framework out of the box contains most if not all widgets for both consistent Android and iOS development in the material and Cupertino packages respectively. Many animations and interactions, such as scroll, are native by default and are performed differently depending on the platform, on which the application is launched. So it is very easy to make the application feel natural on different platforms. Sometimes however behavior needs to be implemented manually. We can use different widgets and animations depending on the platform, so to achieve the maximum native feel some understanding of Android and iOS platforms' design conventions needs to be present, so the cases that are not handled automatically can be implemented manually (often easily with already pre-defined widgets from the corresponding package).

A similar situation is with desktop applications, however, the ability to build Flutter for desktop is fairly new at the time of the article so even though there are already several desktop applications in a production written in Flutter many features remain absent. For our needs, the firebase support has not yet been implemented for the desktop and only MacOS firebase support is in beta now so we do not test those.

For the web application, some changes need to be implemented in the code. Both mobile and web have platform-specific packages, using which on an unsupported platform will lead to an error, so if their usage is required a condition, that checks the platform needs to be used. Apart from that launching an app on the web is easy, however, at the time of the article, the Flutter web is not ideal for every use. The Flutter web feels like a mobile application on the web, even text selection is a fairly recent feature, so there are only the following use cases, when it can be useful:

- Progressive Web Apps;
- Single Page Apps;
- Existing Flutter mobile apps.

Flutter is not suitable for static websites with text-rich flow-based content.

6.2.3. *Firebase setup*

The firebase project is easily created on the corresponding website. It's important to know that by default the project is restricted and you need to connect a bank card to it so its full capabilities, like Cloud Functions, are unlocked. The charges are not made until the specified threshold of usage, since this is a test app and the threshold is quite high we do not need to worry about it.

Next comes the connection of the firebase to our Flutter application. At the time of the article only Android, iOS, and Web Platforms are supported with the MacOS platform in Beta. Each platform needs its separate configuration, but the usage in the Flutter app will be with the same code.

Finally, for us to be able to connect to databases and authentication services we need to set up them. For databases and storage, the server location needs to be chosen, the west Europe multi-regional location has been chosen. For the authentication, we need to choose providers, the email/password and google has been chosen.

6.2.4. *Authentication*

For every firebase service, a corresponding official Flutter package has been made. For the authentication the package is `firebase_auth`. This package handles the whole authentication process, and user caching and provides a stream of authentication state changes. It also supports password reset, where an email, defined in the Firebase console, will be sent to the user.

The usage of this package is simple, for email/password authentication only a simple method with email and password needs to be sent, if the user does not exist he is created automatically. To store the user's info a Cloud Firestore needs to be used, the user's document has the same id as returned from the authentication method.

For the google sign in an additional package `google_sign_in` is required, it provides a native google authentication process and returns necessary credentials, that are later used with the firebase authentication.

6.2.5. *Cloud Firestore, Storage and Real-time database usage*

For the cloud firestore the `cloud_firestore` package is used. To gain access to the document or collection the path can be inputted as a string or built with a builder-like pattern. Cloud firestore support web socket out of the box, so access to any document or collection of documents can be responsive to changes, reflecting any changes in the database in the application without the need for data refresh requests. It makes the development process faster and more efficient since any change to the document is reflected automatically, without the need to handle such cases manually.

Such a responsive approach has a disadvantage in the case of data pagination. When we request a page of data using a web socket loading the next page can be done in one of two ways:

1. creating a new connection to the next page of data and saving it separately. Such an approach makes it burdensome to handle all of the open connections and also it can handle items addition and removal incorrectly;

2. disposing of a previous connection and creating a new one with the data amount restriction increasing by a one-page count. Such an approach works perfectly and has no incorrect behavior, however, it re-requests previously loaded pages of data again, and any operations on documents are charged.

So, in terms of cost-effectiveness, it would be better to request data statically and reflect its changes only with a refresh sequence. In that case, we trade of user experience for the lowering cost of the system to run. The decision of whether the data should be loaded using web sockets or an HTTP request should be decided for each situation separately.

The security rules for the operations with a Cloud Firestore database are made in a JavaScript-like language [12]. To add a rule, you add a path to the document you want to add a rule to (for example `"/users/{userId}"`) and define conditions for read and write operations. The rules use the authenticated user's id so you need to use the Firebase authentication service to able to establish flexible security.

Cloud Storage works in a similar way to the Cloud Firestore, only access to the files is made through the public URL, which is acquired with a special method of the corresponding package – `firebase_storage`. The URL will grant complete access to the file that it leads to. The rules, which are defined the same way as for Cloud Storage, only affect the method, that returns a public URL for the file, so if, for example, you need to fetch multiple pictures to show the user you would need to call the URL method multiple times for each file for the rules to be applied. That is an ineffective and long process, so saving the URL of the file in the corresponding document and generating this URL with the file upload would be a more effective,

scalable approach. If a file needs to be protected it would be better to encrypt the URL than use the first method.

A real-time database (RTDB) is a predecessor of a Cloud Firestore. It's an efficient, low-latency solution that requires synced states across clients in real-time [13]. Cloud Firestore scales better and has richer and faster queries, so using it would be better to use it instead of RTDB. The only viable use for an RTDB that has been implemented in the test application is controlling user presence. In cooperation with cloud functions, we can track when the user is online since unlike Cloud Firestore RTDB has an `onDisconnect` option. When the user's internet connection disappears the cloud function changes the user's state in the corresponding document, allowing us to know the user's presence. That can be used, for example, to know if we need to send the user a push notification for, for example, a new message in a chat. If the user is already present in a chat the notification should not be sent.

6.2.6. Cloud Functions

The Cloud Functions are a convenient way of adding side effects to operations via triggers and creating HTTP endpoints if needed. For example, we can send a push notification on document creation without the need to perform this operation by hand from the client and spend the user's traffic and time. The triggers for document creation, update, and deletion are available. Also, there is a trigger that is called on all 3 of those events – write trigger.

As to HTTP cloud function, one was used in our case – a webhook for stripe operations. To know if the payment or any other stripe event was completed successfully or with an error a webhook, which is basically an HTTP endpoint, needs to be defined in the Stripe console. This webhook will be called when defined in the Stripe console events occur. In our case, a user's document contains info on an ongoing payment process and the status changes based on the data received in the webhook.

Important to know that Cloud Functions have unrestricted access to all Firebase data and security rules do not apply to operations inside them.

6.2.7. Remote config, In-App Messaging and AdMob

Remote config is just a cloud key-value pairs storage, that lets you change the app's behavior without the need to release an update. For example, the app's theme change depending on the season can be handled that way.

Push notifications have been implemented in the following way: a user's device token is fetched and stored in their document on the authorization. When an event, that needs a push notification to be sent occurs we create a document in a notifications collection in Cloud Firestore with all the needed info for the notification, such as the sender and receiver id and the notification type. The Cloud Functions trigger on notifications document creation uses an In-App messaging API to send a push to the device token of the receiver [14].

For AdMob, the `google_mobile_ads` package is used. The following advertisement types are available:

- banner – a rectangle above or below the screen;
- interstitial – full-screen ads that need to be closed by the user;
- native – flexible type, that allows placing ads wherever the developer wants, so ads are effectively integrated into the app's UI;
- rewarded – an ad that rewards users for watching short videos and interacting with playable ads and surveys.

Advertisements are a great way of monetizing your app as long as they do not severely interfere with the user's experience.

6.2.8. Analytics

Google Analytics, Crashlytics, and Performance are all simple to implement via their corresponding Flutter packages. The only thing needed for them is a Firebase project setup, which has already been done earlier.

Google Analytics allows us to send any kind of event with some simple key-value data with them. For example, when the user enters a screen, we can send a corresponding event with the specific screen data. When a sufficient amount of data is gathered the app usage can be analyzed and development priorities set based on users' experiences [15].

Crashlytics service is Google Analytics for errors and crashes. It tracks both developer-defined errors, that are sent manually through the API and app crashes. The errors then can be seen in the corresponding tab in a Firebase console and dealt with accordingly. In the process of testing this service, it was discovered that there is sometimes not enough info on an error or crash, which makes fixing it hard, so maybe it would be better to use a separate service. Also, Crashlytics does not allow for user reports, which is a minus.

Performance is as simple as sending a request on some operation start, and another one on its end. Useful for complex operations to check their performance on many different users' devices.

6.2.9. App distribution and web hosting

App distribution allows to conveniently distribute of mobile iOS and Android applications without the need to go through their respective console setups. You can invite testers through their emails so that they can gain access to the application. The advantages of using Firebase App Distribution compared to platforms' consoles are the following:

- managing both iOS and Android pre-release distributions from the same place;
- early releases can be delivered to testers quickly, with fast onboarding, no SDK to install, and instant app delivery;
- combined with Crashlytics we can get insights into the stability of test distributions.

Hosting your web application can be done on a custom domain, which is usually bought, or on a domain provided by Firebase. Firebase also allows the creation of Beta channels, allowing you to host test applications on a temporary domain. The hosting needs some simple console application to set up your project after which you can easily deploy your application with a command.

Conclusions

As a result of the study, research has been conducted on Flutter and Firebase technologies and their synergy.

Flutter has been tested for the satisfaction of modern application development standards and the speed and ease of the development cycle.

A test service has been developed with the client side on Flutter and Firebase as a backend. Most Firebase services useful in a typical service have been used. It required 1 developer with only Flutter expertise to develop in a short period of time a fully functioning service with most typical features, that can be distributed for iOS, Android, and Web platforms.

In conclusion, a Flutter and Firebase combination has been proven a very cost-effective approach for developing client-server applications, especially for simple ones, compared to a more conventional approach with multiple native and server applications.

References

- [1] *Vaibhav Patil*. Flutter-Modern and Easy Technology to Build Applications // ResearchGate. – 2023. – Vol. 11, No. 2, P. 458-459.
- [2] *Artem Velykyy*. FLUTTER: A FULL INTRODUCTION TO THE FRAMEWORK // Axon. – 2020. – P. 2-3.
- [3] Flutter architectural overview [Electronic resource] – Resource access mode: <https://docs.flutter.dev/resources/architectural-overview>.
- [4] *Damian Bialkowski, Felipe Diniz Dallilo*. FLUTTER UM FRAMEWORK PARA DESENVOLVIMENTO MOBILE // ResearchGate. – 2022. – P. 3-5.
- [5] *Slavimir Stošović, Dušan Stefanović, Milan Bogdanović, Nikola Vukotić*. THE USE OF THE FLUTTER FRAMEWORK IN THE DEVELOPMENT PROCESS OF HYBRID MOBILE APPLICATIONS // ResearchGate. – 2022. – P. 5-6.
- [6] *Anil Trimbakrao Gaikwad*. FIREBASE – OVERVIEW AND USAGE // ResearchGate. – 2022. – P. 2-4.
- [7] *Jashandeep Singh, Swapnil Srivastva, Dipanshu Raj, Shubhampreet Singh, Mir Junaid Rasool*. FLUTTER AND FIREBASE MAKING CROSS-PLATFORM APPLICATION DEVELOPMENT HASSLE-FREE // IRJMETS. – 2022. – Vol. 4, No. 4, P. 3-5.
- [8] *Damian Bialkowski, Jakub Smolka*. Evaluation of Flutter framework time efficiency in context of user interface tasks // ResearchGate. – 2022. – P. 4-5.
- [9] *Omar Almootassem, Syed Hamza Husain, Denesh Parthipan, Qusay H. Mahmoud*. A Cloud-based Service for Real-Time Performance Evaluation of NoSQL Databases // Arxiv. – 2017. – P. 3-4.
- [10] *Azad, Avi Chaudhary, Jatin Chauhan, Basant Soam, Mr. Ashwini Kumar*. ANDROID APPLICATION USING FLUTTER AND FIREBASE WITH LBRS TO FIND PEOPLE OF THE SAME INTEREST AND COMMUNICATION PLATFORM // IRJMETS. – 2022. – Vol. 4, No. 5, P. 4773-4775.
- [11] *Wisal Khan, Teerath Kumar, Zhang Cheng, Kislay Raj, Arunabha M Roy, Bin Luo*. SQL and NoSQL Databases Software architectures performance analysis and assessments – A Systematic Literature review // 3. – 2022. – P. 3.
- [12] Structuring Cloud Firestore Security Rules [Electronic resource] – Resource access mode: <https://firebase.google.com/docs/firestore/security/rules-structure>.
- [13] *Shayan Bagchi*. Firebase-A Cloud Hosted NoSQL Database // ResearchGate. – 2022. – P. 8-9.
- [14] *Bhavin M. Mehta, Nishay Madhani, Radhika Patwardhan*. Firebase: A Platform for your Web and Mobile Applications // IJARSE. – 2017. – Vol. 6, No. 4, P. 46-47.
- [15] *Julian Harty, Haonan Zhang, Lili Wei, Luca Pascarella, Mauricio Aniche, Weiyi Shang*. Logging Practices with Mobile Analytics: An Empirical Study on Firebase // Arxiv. – 2021. – P. 1-3.

The article was delivered to editorial staff on the 27.05.2022

Терещенко Гліб Юрійович¹, Хімич Єгор Русланович²¹ХНУРЕ, м. Харків, Україна, hlib.tereshchenko@nure.ua, ORCID iD: 0000-0001-8731-2135²ХНУРЕ, м. Харків, Україна, yehor.khimich@nure.ua

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТРАДИЦІЙНИХ ТА ГІБРИДНИХ МОДЕЛЕЙ СХОВИЩ ЗОБРАЖЕНЬ У СФЕРІ ВЕЛИКИХ ДАНИХ

Було проведено аналіз традиційних та гібридних моделей сховищ зображень у сфері великих даних, зроблено детальне порівняння моделей зберігання зображень в контексті великих обсягів даних. Дослідження аналізує ключові метрики продуктивності, такі як швидкодія, потужність та масштабованість, для обох типів моделей. Було виокремлено переваги та обмеження кожного підходу і розглянуто їхні особливості в контексті використання у великих проєктах зберігання зображень. Додатково, зроблено акцент на областях застосування кожної моделі і надано рекомендації щодо вибору оптимального рішення для конкретного сценарію великих даних.

ВЕЛИКІ ДАНІ, ЗОБРАЖЕННЯ, ТРАДИЦІЙНА МОДЕЛЬ, ГІБРИДНА МОДЕЛЬ, СХОВИЩЕ ДАНИХ, ПРОДУКТИВНІСТЬ, ЕФЕКТИВНІСТЬ

Tereshchenko G. Yu., Khimich Y. R. Comparative analysis of traditional and hybrid image repository models in the domain of big data. An analysis of traditional and hybrid models of image repositories in the field of big data was conducted, providing a detailed comparison of image storage models in the context of large data volumes. The study examines key performance metrics such as speed, power, and scalability for both types of models. The advantages and limitations of each approach were identified, considering their specific characteristics in the context of image storage in large-scale projects. Additionally, emphasis was placed on the application domains of each model, and recommendations were provided for selecting the optimal solution for specific big data scenarios.

BIG DATA, IMAGES, TRADITIONAL MODEL, HYBRID MODEL, DATA WAREHOUSE, PRODUCTIVITY, EFFICIENCY

Вступ

Великі дані (Big Data) є однією з найбільш актуальних та швидкозростаючих галузей сучасного інформаційного суспільства. Завдяки постійному зростанню обсягів даних, виникає потреба у розробці та вдосконаленні ефективних моделей сховищ даних, зокрема сховищ зображень.

У цій роботі було проведено порівняльний аналіз традиційних та гібридних моделей сховищ зображень у сфері великих даних. Традиційні моделі, які базуються на реляційних базах даних, вже довгий час використовуються для зберігання та управління даними. Однак, з появою великих обсягів зображень, традиційні моделі можуть стикатися з проблемами швидкодії та масштабованості.

Гібридні моделі сховищ зображень, з іншого боку, поєднують переваги реляційних баз даних та багатовимірних баз даних. Вони використовують гібридні сервери аналітичної онлайн обробки, які забезпечують швидкість вбудовування попередньо сформульованої структурної інформації та зберігання інформації по розподілу даних. Це дозволяє ефективно зберігати та обробляти великі обсяги зображень, забезпечуючи мінімізацію помилок та швидкість генерації результуючих звітів.

Порівняльний аналіз традиційних та гібридних моделей сховищ зображень у сфері великих даних є важливим кроком у розвитку сфери великих даних та забезпеченні ефективного управління зображеннями. Розуміння переваг та обмежень традиційних та гібридних моделей сховищ зображень дозволить

організаціям та дослідникам зробити обґрунтований вибір при розробці та впровадженні систем зберігання та аналізу зображень.

У наступних розділах було детально розглянуто основні характеристики та особливості традиційних та гібридних моделей сховищ зображень, проведено порівняльний аналіз їх ефективності та розглянуто приклади використання в реальних сценаріях.

1. Огляд традиційних моделей сховищ зображень у сфері великих даних

1.1. Визначення та особливості традиційних моделей сховищ зображень

Зображення є невід'ємною частиною багатьох додатків і систем, включаючи сферу великих даних. Для ефективного зберігання та управління великим обсягом зображень часто використовуються традиційні моделі сховищ. Було розглянуто загальні принципи зберігання зображень у таких моделях та основні компоненти, що входять до складу традиційного сховища зображень.

Традиційні моделі сховищ зображень базуються на використанні реляційних баз даних для зберігання та управління зображеннями. Основним принципом зберігання зображень у таких моделях є використання VLOB-об'єктів (Binary Large Objects). Кожне зображення зберігається у вигляді бінарних даних, які можуть містити значну кількість інформації.

Одним з основних завдань традиційного сховища зображень є забезпечення ефективного зберігання та розміщення зображень у базі даних. Для цього

використовуються спеціальні типи даних, такі як BLOB або VARBINARY, які дозволяють зберігати бінарні дані без втрати інформації.

Традиційне сховище зображень складається з декількох основних компонентів, які спільно забезпечують ефективне зберігання та управління зображеннями:

- У реляційних базах даних, зображення зберігаються у відповідній таблиці, де кожен запис відповідає одному зображенню. Колонка типу BLOB або VARBINARY використовується для збереження фактичних бінарних даних зображення. Це дозволяє ефективно зберігати та відтворювати зображення без втрати інформації.

- Поміж бінарних даних зображень, традиційні моделі сховищ зображень також дозволяють зберігати метадані, які описують зображення. Ці метадані можуть включати атрибути, такі як назва зображення, розмір, формат файлу, дата створення та інші відомості, які допомагають ідентифікувати та класифікувати зображення.

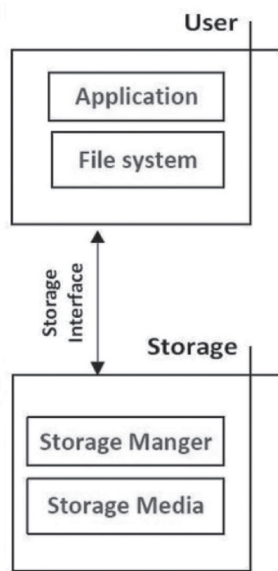


Рис. 1. Дизайн традиційної моделі сховищ

У традиційних моделях сховищ зображень, таблиці зображень можуть бути пов'язані з іншими таблицями у базі даних. Наприклад, може бути встановлений зв'язок з таблицею користувачів, щоб встановлювати власників зображень або з таблицею метаданих для додаткового опису та класифікації зображення.

1.2. Аналіз архітектури традиційних моделей сховищ зображень

Централізована архітектура є однією з найпоширеніших для традиційних моделей сховищ зображень. У цій архітектурі всі зображення зберігаються в одному центральному сховищі даних. Це спрощує управління та контроль над зображеннями, оскільки всі дані зосереджені в одному місці. Однак, ця архітектура може стати обмеженням при обробці великого

обсягу зображень, оскільки всі операції виконуються в одному центральному сховищі.

Розподілена архітектура передбачає розподіл зображень між різними сховищами даних або вузлами. У цій архітектурі зображення розбиваються на частини, які розміщуються на різних вузлах або сховищах. Це дозволяє розподілити навантаження та підвищити продуктивність обробки зображень. Однак, розподілена архітектура вимагає додаткових механізмів для керування та синхронізації даних між вузлами, що може бути складним завданням.

Ієрархічна архітектура базується на ієрархічній структурі для зберігання та управління зображеннями. У цій архітектурі зображення організовані у вигляді дерева або графа, де кожен вузол представляє колекцію зображень. Це дозволяє швидкий доступ до конкретних зображень та швидку навігацію по структурі. Ієрархічна архітектура забезпечує гнучкість та масштабованість при роботі з великим обсягом зображень.

Оцінка ефективності та продуктивності архітектури традиційних моделей сховищ зображень відіграє важливу роль при виборі гідного рішення для конкретних потреб. Для оцінки ефективності можуть бути використані такі метрики, як швидкість збереження та витягування зображень, час відклику, масштабованість та надійність системи. Також важливо враховувати розмір сховища, вартість інфраструктури та складність розгортання та обслуговування архітектури.

1.3. Виклики та обмеження традиційних моделей сховищ зображень у сфері великих даних

Розвиток технологій і зростання обсягу зображень у сучасному світі ставлять перед традиційними моделями сховищ зображень ряд викликів та обмежень. Одним з головних викликів при збереженні великого обсягу зображень є обробка та збереження такої великої кількості даних. Традиційні моделі можуть стикатися з обмеженнями щодо пропускну здатності мережі, обсягу сховища та швидкодії операцій зчитування або запису даних.

Зростання обсягу зображень вимагає високої швидкості обробки, щоб забезпечити швидкий доступ до даних та ефективну обробку запитів. Традиційні моделі можуть виявитися недостатньо продуктивними для роботи з великими обсягами зображень, особливо при високих навантаженнях або одночасному доступі багатьох користувачів.

Традиційні моделі сховищ зображень можуть мати обмеження щодо масштабованості, тобто їх здатності працювати з ростом обсягу даних. Деякі моделі можуть бути недостатньо гнучкими або ефективними при розширенні розмірів сховища та збільшенні кількості користувачів.

Також, традиційні моделі можуть стикатися з обмеженнями щодо продуктивності, особливо при обробці великої кількості запитів або складних аналітичних операцій. Відсутність оптимізації та швидкодії може призводити до затримок у відповіді та недосягнення очікуваних рівнів продуктивності.

Традиційні моделі можуть мати обмеження щодо забезпечення доступу до даних, особливо при одночасному доступі багатьох користувачів. Конкуренція за ресурси та висока вимогливість до швидкості доступу можуть призвести до проблем зі шкалуванням та затримками в обробці запитів.

2. Огляд гібридних моделей сховищ зображень у сфері великих даних

2.1. Визначення та особливості гібридних моделей сховищ зображень

Гібридні моделі сховищ зображень є інноваційним підходом до зберігання та управління зображеннями, який комбінує переваги різних технологій. Гібридні моделі сховищ зображень поєднують різні технології та архітектури з метою оптимізації зберігання та управління зображеннями. Вони можуть комбінувати реляційні бази даних з системами зберігання об'єктів, розподілені файлові системи або хмарні сервіси. Основна ідея полягає в тому, щоб використовувати кожну технологію на тому етапі обробки та зберігання зображень, на якому вона найбільш ефективна.

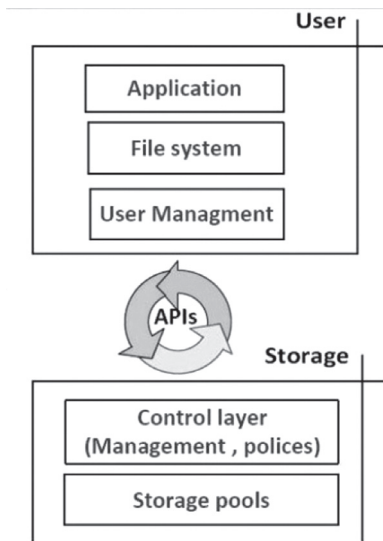


Рис. 2. Дизайн гібридної моделі сховищ

Одним з ключових принципів гібридних моделей є розподіленість та горизонтальне масштабування. Це дозволяє розподілити обробку та зберігання зображень між різними компонентами системи, що призводить до покращення продуктивності та масштабованості.

Гібридні моделі дозволяють оптимізувати зберігання та обробку зображень для різних типів

операцій. Наприклад, реляційні бази даних можуть бути використані для зберігання метаданих та інформації про зображення, тоді як системи зберігання об'єктів можуть бути ефективними для зберігання самого зображення у вигляді об'єкта.

Гібридні моделі забезпечують еластичність та гнучкість в роботі зі зображеннями. Вони дозволяють легко адаптуватися до змінних вимог щодо обсягу даних, продуктивності та доступу до даних. За допомогою гібридних моделей можна розширювати та змінювати систему зберігання зображень, враховуючи змінюються потреби та технологічні розвитки.

Переваги гібридних моделей зберігання зображень:

- Оптимальне використання різних технологій.

Гібридні моделі дозволяють оптимально використовувати різні технології залежно від конкретних потреб та вимог до зображень. Це дозволяє досягти оптимальної ефективності та продуктивності системи.

- Масштабованість.

Гібридні моделі забезпечують масштабованість, що дозволяє збільшувати обсяги даних та кількість користувачів без значного зниження продуктивності та доступності.

- Гнучкість та адаптивність.

Гібридні моделі надають гнучкість та адаптивність в роботі з зображеннями. Вони можуть бути змінені або розширені відповідно до змінних потреб, технологічних нововведень та розвитку інфраструктури.

2.2. Аналіз архітектури гібридних моделей сховищ зображень

Одна з типових архітектур гібридних моделей використовує комбінацію реляційних баз даних та систем зберігання об'єктів. У цій архітектурі, метадані та інформація про зображення зберігаються у реляційній базі даних, тоді як самі зображення зберігаються у вигляді об'єктів у системі зберігання об'єктів. Це дозволяє ефективно керувати метаданими та швидко доступатися до зображень.

Іншою типовою архітектурою гібридних моделей є поєднання розподіленої файлової системи та хмарних сервісів.

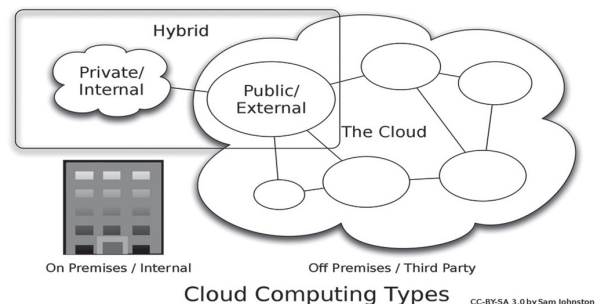


Рис. 3. Схема архітектури гібридних моделей з використанням хмарних технологій

У цій архітектурі, зображення зберігаються у розподіленій файлової системі, яка може бути розподілена по різних серверах. Додатково, для зберігання та обробки великого обсягу зображень можуть використовуватися хмарні сервіси, які надають масштабованість та доступність.

Оцінка продуктивності архітектур гібридних моделей залежить від різних факторів, включаючи швидкість зберігання та витягування зображень, обробку запитів та масштабованість системи. Важливо враховувати особливості кожної архітектури та оптимізувати її для досягнення найкращої продуктивності.

Ефективність архітектур гібридних моделей оцінюється з точки зору використання ресурсів, які вони вимагають для зберігання та обробки зображень. Важливо забезпечити оптимальне використання різних технологій та розподілити завдання між компонентами системи для досягнення ефективності.

2.3. Аналіз архітектури гібридних моделей сховищ зображень

Гібридні моделі дозволяють пристосовуватися до змінних потреб сфери великих даних. Вони можуть бути налаштовані для використання різних архітектур та технологій, що дозволяє враховувати специфіку проекту та вимоги до продуктивності, масштабованості та доступності.

Такі моделі можуть використовувати розподілені системи зберігання, що дозволяє розподіляти дані по різних серверах або хмарних сервісах. Це забезпечує високу доступність, надійність та можливість резервного копіювання даних.

Впровадження гібридних моделей вимагають спеціалізованих знань та розуміння різних технологій. Налаштування та управління комбінованої архітектури можуть бути складними завданнями, які потребують досвіду та експертизи.

Гібридні моделі можуть вимагати значних обчислювальних ресурсів для забезпечення ефективності та продуктивності. Це може включати потужні сервери, мережеві пропускні здатності та інфраструктуру для розподіленого зберігання.

Також, гібридні моделі потребують інтеграції з наявною інфраструктурою та додатками. Це може створювати виклики при забезпеченні сумісності, безпеки та зручного використання з існуючими системами та розробками.

Гібридні моделі сховищ зображень мають багато переваг у порівнянні з традиційними моделями. Вони забезпечують ефективне зберігання та управління великим обсягом зображень, гнучкість та розширюваність, а також розподіленість та резервне копіювання. Однак, впровадження гібридних моделей вимагає уваги до викликів та обмежень, таких як складність налаштування та управління, високі вимоги до обчислювальних ресурсів та інтеграція

з наявною інфраструктурою. Ретельне аналізування потреб і вимог проекту, а також експертна підтримка можуть допомогти здійснити успішне впровадження гібридних моделей сховищ зображень у сфері великих даних.

3. Порівняльний аналіз

3.1. Порівняння ефективності та продуктивності традиційних та гібридних моделей

У сфері зберігання зображень у великих обсягах даних, вибір оптимальної моделі сховища має велике значення для досягнення високої ефективності та продуктивності. Традиційні та гібридні моделі є двома широко використовуваними підходами, і порівняння їх ефективності та продуктивності може надати корисний інсайт для вибору оптимального рішення.

Одним з ключових критеріїв порівняння є швидкодія моделей сховищ зображень. Традиційні моделі, такі як реляційні бази даних, можуть мати обмеження у швидкості доступу до зображень через потребу у виконанні складних операцій пошуку та з'єднання даних. З іншого боку, гібридні моделі, які комбінують реляційні бази даних з розподіленими системами зберігання, можуть забезпечувати більшу швидкодію завдяки розпаралелюванню обробки та розподіленому доступу до даних.

Потужність моделей сховищ зображень визначається їх здатністю обробляти великі обсяги даних та виконувати операції швидко і ефективно. Традиційні моделі можуть бути обмеженими у потужності через обмеження апаратного забезпечення або архітектури бази даних. У той же час, гібридні моделі можуть надати більшу потужність завдяки використанню розподіленої обробки, паралельних обчислень та оптимізованої архітектури для роботи з великими обсягами даних.

Масштабованість є важливою характеристикою моделей сховищ зображень у сфері великих даних. Традиційні моделі можуть зазнавати складнощів при масштабуванні, особливо якщо потрібно обробляти великі обсяги зображень. Гібридні моделі можуть бути більш масштабованими завдяки розподіленій архітектурі, можливості горизонтального масштабування та узгодженій обробці даних у розподілених середовищах.

Оцінка ефективності рішень зберігання зображень може базуватися на різних критеріях, таких як продуктивність, надійність та гнучкість.

Продуктивність вимірюється швидкістю та завантажувальною здатністю системи сховища зображень. Традиційні моделі можуть мати обмежену продуктивність через архітектурні обмеження або операції вводу-виводу зображень. Гібридні моделі можуть надавати більшу продуктивність, оптимізуючи операції

вводу-виводу та використовуючи розподілені системи зберігання.

Надійність відображає стійкість та безпеку даних у моделях сховищ зображень. Традиційні моделі можуть мати обмеження щодо резервного копіювання та відновлення даних, що може призвести до втрати зображень у разі виникнення аварійної ситуації. Гібридні моделі можуть забезпечувати вищий рівень надійності через резервне копіювання, реплікацію даних та інші механізми забезпечення доступності.

Гнучкість відображає можливість моделі сховища зображень працювати з різними типами зображень та інтегруватися з іншими системами. Традиційні моделі можуть бути обмеженими у гнучкості через обмежену підтримку форматів зображень або обмежені можливості інтеграції. Гібридні моделі можуть бути більш гнучкими, забезпечуючи розширені можливості обробки різних форматів зображень та легку інтеграцію з іншими системами.

3.2. Аналіз викликів та переваг традиційних та гібридних моделей

У сфері зберігання зображень у великих обсягах даних, традиційні та гібридні моделі мають свої виклики, обмеження та переваги.

Одним із викликів традиційних моделей є можливість стикання з обмеженнями масштабованості при роботі з великими обсягами зображень. Обробка та зберігання великих даних може вимагати значних обчислювальних ресурсів та інфраструктури.

Традиційні моделі можуть мати обмежену продуктивність у забезпеченні швидкодії та завантажувальної здатності. Операції вводу-виводу зображень та обробки даних можуть стати недостатньо ефективними для потреб великих обсягів даних.

Ще одним викликом таких моделей можуть бути обмеження у гнучкості підтримки різних форматів зображень та інтеграції з іншими системами. Вони можуть не забезпечувати достатньої гнучкості для роботи з різноманітними типами даних та розширення функціональності.

З іншого боку, традиційні моделі зберігання зображень можуть бути простішими у реалізації та використанні. Вони можуть мати меншу складність у розгортанні та управлінні.

Вони також можуть бути вартісними, оскільки вони використовують стандартне обладнання та програмне забезпечення, яке може бути доступним із комерційних джерел.

Традиційні моделі мають широке застосування та підтримку у великій кількості проєктів. Вони можуть бути стабільними та надійними рішеннями для зберігання зображень.

Виклики гібридних моделей можуть проявлятися у складності реалізації порівняно з традиційними

моделями. Вони можуть вимагати розробки спеціалізованих алгоритмів та розподіленої інфраструктури для досягнення потрібної продуктивності та масштабованості. Також, гібридні моделі можуть вимагати складнішого управління через розподілену архітектуру та розсіяність даних. Координація та синхронізація можуть бути викликами для забезпечення цілісності та доступності даних.

Гібридні моделі можуть бути вартісними внаслідок потреби у спеціалізованій інфраструктурі та розподіленій обробці даних. Вартість розгортання та управління може бути вищою порівняно з традиційними моделями.

Але, гібридні моделі також мають переваги перед традиційними – вони можуть надати більшу швидкоддю завдяки розподіленій архітектурі та розпаралюванню обробки даних. Це особливо корисно при операціях обробки великих обсягів зображень.

Гібридні моделі можуть мати більшу потужність у виконанні складних обчислювальних завдань, таких як виявлення об'єктів, розпізнавання образів тощо. Вони можуть використовувати потужні графічні прискорювачі та спеціалізоване обладнання для покращення продуктивності. Вони можуть бути більш гнучкими в підтримці різних форматів зображень та інтеграції з іншими системами. Вони можуть надати розширені можливості для обробки та аналізу зображень з використанням різноманітних алгоритмів та інструментів.

Порівнюючи обидва типи моделей, можна зрозуміти, що традиційні моделі мають свої переваги в простоті використання та стабільності, а гібридні моделі - у швидкодії, продуктивності та гнучкості. Вибір між цими моделями залежить від конкретних потреб проєкту та вимог до продуктивності, масштабованості та функціональності системи зберігання зображень.

Висновки

У результаті порівняльного аналізу традиційних та гібридних моделей сховищ зображень у сфері великих даних було зроблено наступні загальні висновки щодо ефективності та використанні обох типів моделей.

Ефективність: гібридні моделі сховищ зображень виявляються більш ефективними у порівнянні з традиційними моделями. Вони поєднують переваги різних підходів, забезпечуючи оптимальне поєднання швидкодії, масштабованості та продуктивності. Гібридні моделі можуть забезпечити більшу гнучкість та ефективну роботу з великими обсягами зображень.

Використання: вибір між традиційними та гібридними моделями залежить від конкретних потреб проєкту та вимог до продуктивності, масштабованості та функціональності системи зберігання зображень.

Традиційні моделі можуть бути виправданими в простих сценаріях, де вимоги до обробки та зберігання зображень є обмеженими. Гібридні моделі рекомендуються у випадках, коли потрібна більша гнучкість, масштабованість та оптимальна продуктивність у роботі з великими обсягами зображень.

Однак, при використанні гібридних моделей необхідно враховувати певні виклики та обмеження, такі як складність розробки та налагодження, вищі вимоги до обладнання та інфраструктури, а також необхідність забезпечення інтеграції різних компонентів системи.

Остаточний вибір моделі залежить від унікальних вимог проекту, доступних ресурсів та потреб користувачів. Рекомендується провести детальний аналіз вимог, враховуючи масштаби проекту, доступні ресурси, потреби користувачів та наявні навички розробників, щоб визначити найбільш підходящу модель зберігання зображень у сфері великих даних. При цьому варто враховувати переваги гібридних моделей щодо ефективності та масштабованості, а також потенційні виклики та затрати, пов'язані з їх впровадженням.

Список літератури:

- [1] *Smith, J., & Johnson, A.* (2022). Comparative Study of Traditional Image Storage Models in Big Data Applications // *International Journal of Data Science and Analytics*, 10(3), 245-260.
- [2] *Thompson, R., & Chen, S.* (2021). Hybrid Image Storage Models: A Comprehensive Review and Performance Evaluation // *Journal of Big Data*, 8(1), 78.
- [3] *Lee, H., Park, S., & Kim, J.* (2023). Performance Analysis of Traditional and Hybrid Image Storage Architectures in Large-Scale Data Environments // *IEEE Transactions on Big Data*, 9(1), 125-138.
- [4] *Gupta, R., & Singh, M.* (2022). Challenges and Limitations of Traditional Image Storage Models in the Era of Big Data. *International // Journal of Computer Science and Information Technology*, 14(2), 45-57.
- [5] *Patel, A., & Sharma, P.* (2021). Hybrid Image Storage Models: Advantages and Key Features for Big Data Applications // *Journal of Information Science and Technology*, 13(4), 87-102.
- [6] *Brown, K., & Wilson, M.* (2023). Comparative Analysis of Efficiency and Performance of Traditional and Hybrid Image Storage Architectures // *Journal of Big Data Analytics*, 5(2), 70-85.
- [7] *Wang, Y., Zhang, L., & Chen, X.* (2022). Hybrid Image Storage Models: Bridging the Gap between Traditional and Modern Data Management Techniques // *International Journal of Image Processing*, 19(3), 180-195.
- [8] *Park, J., Lee, S., & Kim, H.* (2021). Scalability Challenges of Traditional Image Storage Models in Big Data Environments // *Journal of Data Engineering and Management*, 8(4), 120-135.
- [9] *Kumar, V., & Sharma, A.* (2022). Hybrid Image Storage Models: A Comparative Performance Evaluation in Big Data Applications // *International Journal of Advanced Research in Computer Science*, 13(2), 60-75.
- [10] *Rodriguez, M., & Hernandez, G.* (2023). Challenges and Limitations of Hybrid Image Storage Models in Large-Scale Data Analysis // *Big Data Management*, 7(1), 32-47.
- [11] *Chen, Y., Wu, Q., & Li, M.* (2022). Efficiency Evaluation of Traditional and Hybrid Image Storage Architectures in the Context of Big Data // *International Journal of Computational Intelligence and Applications*, 20(4), 215-230.
- [12] *Gupta, A., & Kapoor, S.* (2021). Hybrid Image Storage Models: Key Architectures and Their Performance Analysis in Big Data Systems // *Journal of Information Systems and Technology Management*, 9(3), 75-90.
- [13] *Kim, H., Park, S., & Lee, J.* (2023). Performance Comparison of Traditional and Hybrid Image Storage Models in Big Data Applications // *International Journal of Database Theory and Application*, 26(1), 58-73.
- [14] *Patel, R., & Shah, K.* (2022). Challenges and Benefits of Hybrid Image Storage Models in the Domain of Big Data Analytics // *International Journal of Data Science and Knowledge Discovery*, 10(4), 89-104.
- [15] *Chen, Z., Wang, X., & Li, J.* (2021). Comparative Evaluation of Efficiency and Productivity of Traditional and Hybrid Image Storage Models in Big Data Environments // *Journal of Data Science and Information Technology*, 14(1), 32-47.

Надійшла до редколегії 15.06 2022



Iona Revenchuk¹, Vladyslav Steshko²

¹ Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
ilona.revenchuk@nure.ua, ORCID: 0000-0002-5188-9538

² Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
vladyslav.steshko@nure.ua

ARCHITECTURAL SOLUTIONS AND OPTIMIZATION METHODS TO IMPROVE THE PERFORMANCE OF NODE.JS AND VUE.JS APPLICATIONS

Software architecture includes a number of important decisions about the organization of a software system, including the choice of structural elements and their interfaces that make up and unite the system into a single whole; the behavior provided by the joint work of these elements; the organization of these structural and behavioral elements into larger subsystems, as well as the architectural style that this organization adheres to. The performance of a web application is an objective measurement and user experience associated with the loading and operation of the program. Performance is about how long a web application takes to load, becomes interactive and responsive, and how smoothly the interaction with content takes place. The architecture and the steps that need to be taken to optimize the program have always been and will be relevant. Any application has its own architecture, but not every application adheres to the rules for building a good architecture, the same applies to optimization. A program with a good architecture is easier to extend and change, as well as to test, configure and understand. As practice shows, people do not like to wait, and even a three-second delay can force a user to close a tab with a slow resource. Therefore, using a number of optimization methods to improve performance will lead to increased usability and will not force the user to leave the resource due to the fact that it is running slowly. Search engines also pay attention to hundreds of parameters when ranking pages in the search. And one of the most important is the speed of data transfer from the server to the client.

ARCHITECTURE, OPTIMIZATION METHODS, PERFORMANCE, WEB SYSTEM, NODE JS, VUE

І.А. Ревенчук, В.Ю. Шешко. Архітектурні рішення і методи оптимізації для підвищення продуктивності додатків на Node.js та Vue.js. Архітектура програмного забезпечення містить у собі низку важливих рішень про організацію програмної системи, серед яких, вибір структурних елементів та їх інтерфейсів, що становлять і об'єднують систему в єдине ціле; поведінка, що забезпечується спільною роботою цих елементів; організацію цих структурних та поведінкових елементів у більшій підсистемі, а також архітектурний стиль, якого дотримується ця організація. Продуктивність веб-додатку – це об'єктивні вимірювання та відчуття користувача, пов'язані із завантаженням і роботою програми. Продуктивність – це про те, як довго веб-додаток завантажується, стає інтерактивним та чуйним, про те, як плавно відбувається взаємодія з контентом. Архітектура та кроки, які необхідно зробити для оптимізації додатку, завжди були і будуть актуальними. Будь-який додаток, має свою архітектуру, але далеко не кожен додаток, дотримується правил, щодо побудови гарної архітектури, те саме стосується і оптимізації. Програму з гарною архітектурою легше розширювати та змінювати, а також тестувати, налагоджувати та розуміти. Як показує практика, люди не люблять чекати і навіть трьох секундна затримка може змусити користувача закрити вкладку з повільним ресурсом. Тому застосування низки методів оптимізації для підвищення продуктивності, призведе до підвищення зручності використання і не змусить користувача залишити ресурс через те, що він повільний. Також пошукові системи при ранжируванні сторінок у пошуку звертають увагу на сотні параметрів. І один із найважливіших – швидкість передачі даних від сервера клієнту.

АРХІТЕКТУРА, МЕТОДИ ОПТИМІЗАЦІЇ, ПРОДУКТИВНІСТЬ, ВЕБ-СИСТЕМА, NODE JS, VUE.

Introduction

At the current stage of World-wide-web development, it is important to improve the performance of a web application, scalability and reliability, by building a correct architecture and using optimization methods.

Technology selection: Node.js and Vue.js, for research, is determined by personal preferences. These technologies are also quite popular, both among companies and developers.

Behind any good product, there is a high-quality architecture and the work done is related to its optimization. Each application may, if not immediately, then after some time, encounter performance-related problems. These problems occur in the following situations:

– the app grows and becomes larger (hundreds of custom scripts, dozens of screens, etc.);

– the application begins to operate with a large amount of data;

– a large number of users appear.

All this can lead to a loss of performance, especially if the application is not optimized and, as a result, users will prefer to use a competitor's product. The use of a poor-quality application architecture negatively affects its further scaling and, as a result, the loss of understanding of the relationship between system components. Maintaining such an application becomes very difficult. Therefore, designing an architecture and paying attention to optimization is the key to a good product.

1. Basic theoretical information about architectural solutions

Software architecture is a set of approaches for organizing a software and hardware complex. Description of system components and relationships between them. Architecture includes approaches, constraints, rules, and heuristics to follow when writing code [1].

Good architecture helps to design and develop a system so that it is easier and more convenient to extend and change it. Therefore:

- if communication between modules is regulated, it is easier to replace their implementation with another one;
- if communication with the outside world is regulated, then there is less chance of data leakage;
- if the code is separated correctly, the application is easier to test;
- if the code is clearly organized, it takes less time to add new features and search for bugs in old ones;
- if the architecture is widely known, immersion in the project is faster [1].

A web app architecture presents a layout with all the software components (such as databases, applications and middleware) and how they interact with each other.

Typically a web-based application architecture comprises 3 core components (3-tier):

- client-side (Presentation layer / Client Layer) – is the key component that interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well, if required.
- server-side (Application Layer / Business Logic Layer) – handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations. It can run and oversee requests from a wide variety of clients.

- database server (Data Layer) provides the required data for the application. It handles data-related tasks. In a multi-tiered architecture, database servers can manage business logic with the help of stored procedures.

It is possible to conditionally divide architectural solutions according to their goals and scope.

Some approaches distribute responsibility between modules. They determine which modules will be responsible for what. These approaches are called architectural patterns, namely: MVC, MVVM, MVP.

MVC (Model-View-Controller) — a scheme for dividing program data and control logic into three separate components: model, view, and controller — so that each component can be modified independently:

- the model provides data and responds to controller commands by changing its state;
- the view is responsible for displaying user model data, responding to model changes.
- the controller interprets user actions, notifying the model of the need for changes.

The pattern is shown in fig. 1.

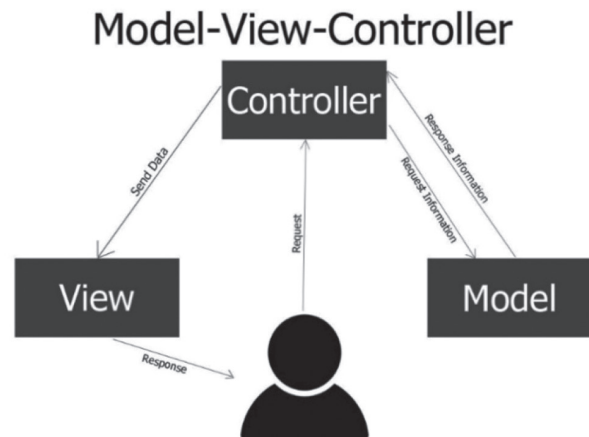


Fig. 1. MVC

MVVM (Model-View-ViewModel) — allows to separate the application logic from the visual part (View). This pattern is architectural, that is, it defines the overall architecture of the application. MVVM consists of three components: Model, ViewModel, and View. The pattern is shown in fig. 2.

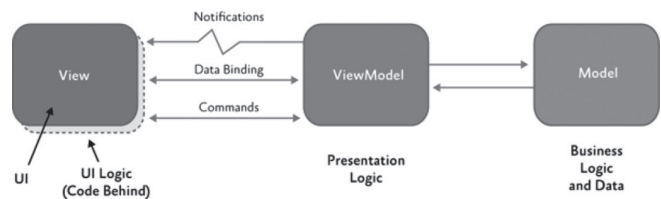


Fig. 2. MVVM

MVP (Model-View-Presenter) is a user interface development pattern. The MVP pattern is derived from MVC, but has a slightly different approach. The main difference is that presenter is not closely related to the model. The presenter takes the place of the controller. The MVP pattern is shown in fig. 3.

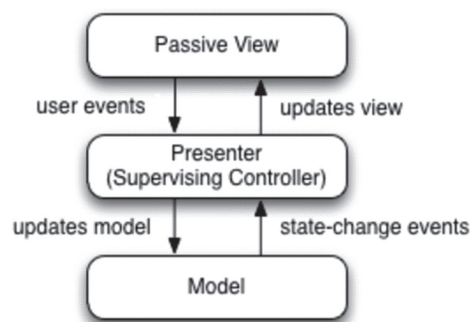


Fig. 3. MVP

The presenter takes over all the logic of data processing, updating the view, and processing user commands. The view in this case is passive: it does nothing but display data as the presenter tells it to. If in MVC the view could take over the formatting of the output, then in MVP the presenter is also responsible for this.

Others determine how close each of the modules is to business logic. For such approaches, it is important which part of the code directly deals with the application task, and which part deals with infrastructure tasks. For example, in a photo processing application, the business logic would be filter functions, and the infrastructure tasks would be calls to the phone’s camera API [2].

Still others manage data flows in the app. They define how modules communicate with each other: directly, indirectly, or using special services such as the event bus. In general, data flows can be organized in a huge number of ways, but most often in practice there are two types in frontend: unidirectional and bidirectional. In a unidirectional data stream, each part of the application can receive or transmit data from another part. The direction of such flow does not change. Data in a bidirectional stream can be transmitted between parts of the application in both directions. This is most often used to link the model and view, so that updating, for example, text in the input field immediately updates data in the model — this is called bidirectional data binding. Frameworks that use bidirectional binding are often reactive — that is, they apply changes instantly not only to the UI, but also to the data being computed. Vue is one such framework.

The following approaches determine the layout of the application. It will be one large program (monolith) or a set of several smaller programs (microservices). In general, there are three types of web application architecture: monolith, microservices and serverless.

A monolith is an architectural solution in which all components and modules are closely related and depend on each other [3]. An example of the monolith architecture is shown in fig. 4.

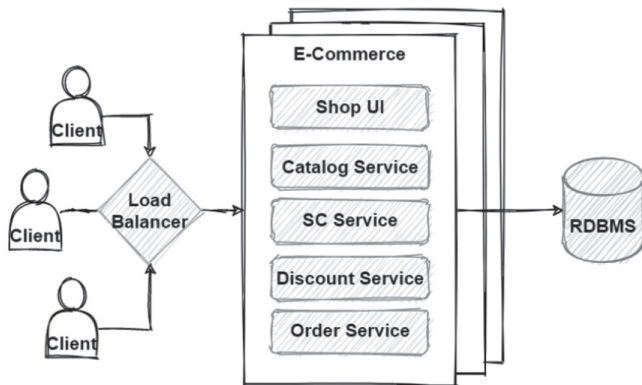


Fig. 4. Monolithic architecture

Monolithic applications work fairly efficiently, as long as they don’t get too big and cause scaling problems.

Microservices is an architectural solution based on the distribution of modules into separate systems that communicate with each other using messages [3]. An example of the architecture is shown in fig. 5.

Microservices architecture solves several challenges that are encountered in a monolithic environment.

In a microservice architecture, the code is developed as loosely-coupled, independent services. Each microservice contains its own database and operates a specific business logic which means you can develop and deploy independent services with ease. Since it’s loosely coupled, microservice architecture provides the flexibility to update/modify and scale independent services. Development becomes easy and efficient and continuous delivery is enabled. For highly scalable and complex applications, microservices is a good choice.

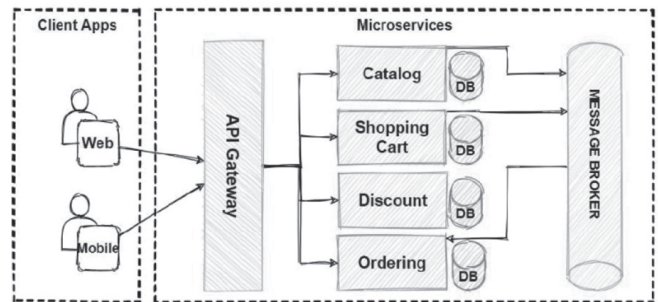


Fig. 5. Microservices architecture

Serverless is an architectural solution that focuses on development, instead of deployment and interaction between services [3]. It is an alternative to microservices that automates the entire deployment thanks to cloud technologies (fig. 6).

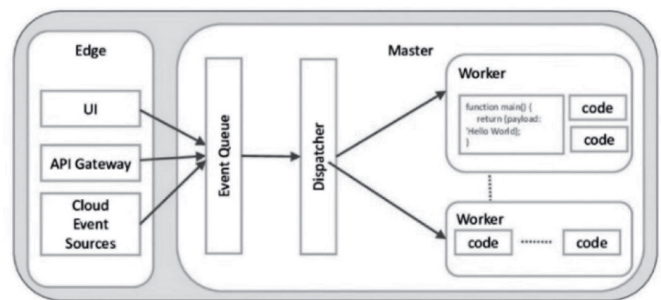


Fig. 6. Serverless architecture

Serverless computing lowers costs as resources are only used when the application is in execution. The scaling tasks are handled by the cloud provider. Moreover, backend code gets simplified. It reduces development efforts, costs and brings faster time to market.

All these architectures are top-level and each is good in its field. To implement the application, you can choose any of the architectures listed above, but it can be ineffective either in terms of development or in terms of operation. Therefore, it is necessary to rely on the application requirements and on the advantages and disadvantages of each of these architectures.

2. Web application performance issues

At the moment, performance problems are among the most common. Their solution requires compliance with a number of rules formed as a result of studying the construction of web applications. Globally, web application

performance issues can be divided into two categories: data transfer and runtime.

- data transfer — loading any resources necessary for the application to work;
- runtime — operation of the application, rendering and processing user input.

Each of these categories contains nuances that distinguish high-quality programs from low-quality ones. Most popular web application performance metrics:

- TTFB — time to receive the first byte;
- FCP — time until the first content display;
- FMP — time to the first significant display;
- TTI — time to start interactivity of page elements.

Among other things, can also identify common causes of performance problems:

- poorly written code can lead to many problems with web applications, including inefficient algorithms, memory leaks, and application deadlocks. Older software versions or integrated legacy systems can also reduce performance;

- non-optimized databases. An optimized database provides the highest level of security and performance, while an unoptimized database can destroy an application. Missing indexes slow down the execution of SQL queries, which can lead to a decrease in the performance of the entire application;

- DNS, firewall and network connection. DNS queries make up the majority of web traffic. That's why a DNS problem can cause so many problems, preventing visitors from accessing your site and leading to errors, 404 errors, and wrong paths. Similarly, network connectivity and firewall performance are critical to access and performance [4];

- external services. The problem with using external services is that they are out of control;
- slow server;
- poor load distribution. This can lead to an increase in response time due to incorrect distribution of new site visitors between servers [5].

3. Problems of choosing software solutions for Node.js

Node.js is a JavaScript runtime environment based on the V8 engine that can compile JavaScript code into machine code. It is an open source cross-platform environment for developing server and network applications that uses an event-driven, non-blocking I / O model, making it simple and efficient for real-time data-intensive applications running across distributed devices.

Node.js has many frameworks, including: `express.js`, `nest.js`, `koa.js`, `adonis`. Each of the presented solutions has its own characteristics and therefore it becomes difficult to choose a suitable solution.

2.1 *Express.js*

`Express.js` is a flexible and minimalistic application framework. It's not built around specific components,

which gives developers the opportunity to experiment. They get lightning-fast customization and a clean JS experience [6].

Features include: fast server-side development; NoSQL database support out of the box; supports MVC architecture; allows developers to create a RESTful API faster.

`Express.js` is suitable for quickly creating web applications and services, as it has API generation tools available.

2.2 *Nest.js*

`Nest` is a server platform built to support developer productivity. `Nest`, a framework written entirely in TypeScript (it also supports JS), is easy to test and includes everything you need.

`Nest`, by default, makes use of the `Express` library under the hood. `Nest` significantly extends its functionality, adds declarativeness, and also helps the developer to build the application according to best architectural practices.

Features include: easily extensible — can be used with other libraries; combines features of functional programming, OOP and functional reactive programming; provides framework APIs that help use various third-party modules available for any platform; has detailed and supported documentation;

`Nest.js` is used to write: clean and reusable code; code using high-level constructs: interceptors, filters, pipes, guards; scalable and tested applications. It provides the right balance of structure and flexibility to effectively manage code.

2.3 *Koa.js*

`Koa.js` is an open source web framework written by the developers of `Express.js`. With `Koa`, they aimed to create a smaller and more reliable platform for web applications and APIs. It offers a wide range of effective methods to speed up the process of creating servers. [7].

Features include: a modern and promising solution; built-in error catchall that prevents crashes; using a context object that contains request and response objects.

Used for creating servers, routes, and handling responses and errors.

2.4 *Adonis.js*

`Adonis` is a Node.js MVC framework for consistent, stable and expressive code. `Adonis` takes care of many of the time-consuming details of development by offering a stable ecosystem for server-side web applications. It was designed to bring joy and ease to developers in their work, so it neglects a consistent and expressive API to develop full-featured web applications. This framework follows the same principles as `Laravel` [8].

Features include: API and session-based authentication system; excellent security system; a powerful ORM that helps create secure SQL queries; validation of user input data.

Ideal for building RESTful APIs.

Each framework has its own capabilities and approaches. Architectural solutions and optimization methods can be applied to each of them. But the most attractive framework is Nest.js. It uses good architectural approaches and, along with optimization methods, will allow to get a scalable, structured and productive backend application.

4. Determination of optimization methods

Common methods for optimizing web applications include the following.

Data transfer

CDN — helps speed up loading for geographically distributed clients.

Resource prioritization — acceleration of page loading using the correct resource loading strategy. Browsers allow you to set priorities for different types of resources and load earlier what is important for the first drawing.

Static compression — is a compression algorithm that reduces the weight of static and, accordingly, increases the download speed.

WebP vs Png vs Jpg. WebP is a great alternative to Png. In addition to the lower weight of the images, WebP is practically the same quality and has a fast loading time [9].

TTFB. Time to receive the first byte of the application page after sending a request from the client. It is an important metric. This is a complex metric that primarily depends on what operations are performed on the server during Request processing. A long response time can be related to dozens of factors: application logic, slow database operation, routing, software platform, libraries, lack of processor power or memory.

HTTP2 can speed up page loading by multiplexing or compressing headers [10].

Runtime

`requestIdleCallback` is a function that allows to execute code at the end of a frame (tick) or when the user is inactive.

`requestAnimationFrame` allows to schedule animations correctly and maximize the chances of rendering at 60 frames per second.

DOM manipulations are expensive, and they need to be performed carefully and meaningfully. Vue, provides optimized work with the DOM due to the use of a lighter version — VirtualDOM.

Virtual scrolling and pagination. These methods are used to display a list with a large number of items. In this case, either infinite loading with virtual scrolling or pagination should be used. If you do not use this approach or use it partially, for example, only infinite loading, then performance can be significantly reduced [11].

60 FPS by pointer-events: none — with this feature you can achieve 60 FPS when scrolling the page. It works on this principle: all mouse handlers are disabled while scrolling.

Build

Webpack is a module bundler. One of the most powerful and flexible tools for building frontend. It analyzes application modules, creates a dependency graph, and then builds the modules in the correct order. It has the ability to optimize the build, which in turn improves performance.

Code splitting — by splitting the code into chunks, you can optimize the first load [12].

Minification — reducing the size of the final build by removing unnecessary characters from HTML, JS, CSS that are not needed to display the page: comments, indents [12].

Dead code — unused code is removed from the final build, thereby speeding up page loading.

If your application uses a library that needs only a few methods, then you need to extract only the methods you need when importing, not all of them. This affects the size of the final bundle and, as a result, performance.

When you download fonts locally, you need to make sure that you are using compressed font formats for modern browsers, such as WOFF and WOFF2 [13].

Lazy loading of modules / routes is a tool that is available in all popular frameworks and libraries. Allows you to “lazily” load chunks of the app’s functionality.

Caching. Cache API, ServiceWorkers, App Shell model. Cache API is a storage for network responses that we have full control over. Service workers are specialized JavaScript assets that act as proxies between web browsers and web servers. They aim to improve reliability by providing offline access, as well as boost page performance. An indispensable aspect of service worker technology is the Cache interface, which is a caching mechanism wholly separate from the HTTP cache. App Shell architecture is one of the most efficient ways of building web apps that are loading almost instantly [14].

Lazy loading of images and videos.

Using indexes.

Application speed testing.

In addition to general methods, there are a number of those that use Vue technology:

– functional components. Let’s assume that there is a simple, small component whose task is to display a particular tag, depending on the passed value. You can optimize this component by adding the functional attribute — functional. A functional component is compiled into a simple function and has no local state.

– keep-alive is a wrapper component over the router. All components in the router will be created and destroyed when switching between routes. If the components are heavy, the interface may hang at the time of switching. Vue provides the ability not to destroy, but to cache and reuse, thereby preserving the state of the component. This optimization will lead to increased memory consumption as Vue needs to keep them alive. This approach should not be applied thoughtlessly [15].

– lazy loading of components. In a traditional component import, the child component is loaded as soon as the browser reaches the import instruction. However, if the component is large, then it makes sense to load it asynchronously. Vue provides this feature out of the box. It will only load the component when needed and display it when it is ready. When using a bundler such as Webpack, the child component will be extracted into a separate file, which will reduce the weight of the file on initial download. Lazy loading can also be applied to components used for routes. Each of the routes will be loaded when requested.

– vuex. Work with commits. Commits, unlike actions, are synchronous. If we assume that there is a need to save a large array of data, then its processing will block the interface during operation. To solve the problem, you can split the array into parts and add them one by one, giving the browser time to display. With this approach, you can add a load indicator, which will improve the UX.

– disabling reactivity. The store contains an array of objects with a high level of nesting, and Vue, according to its behavior, will perform a recursive bypass of all nested fields. If the application is built in such a way that it depends on the top-level nesting object and does not refer to reactive data several levels lower, then this reactivity can be removed, making it easier for Vue to work.

Conclusion

To create apps that people want to use that attract and retain a user, it's necessary to create a good user experience, and part of that experience is fast content loading and responsiveness to user interaction (response time). Therefore, the use of performance improvement methods is an important part of the product being developed. Well-built architecture and good performance are the key to a high-quality product.

References:

- [1] Керівництво Microsoft з проектування архітектури, 2 видання. URL: https://dut.edu.ua/uploads/1_1507_99407341.pdf.
- [2] Роберт Сесіл Мартін, Чиста архітектура – мистецтво розроблення програмного забезпечення // 2019 – 368 с.
- [3] Як обрати архітектуру для Web-додатку. URL: <https://blog.ithillel.ua/articles/web-application-architecture>.
- [4] The 10 most common web app performance problems. URL: <https://www.tricentis.com/blog/10-most-common-web-app-performance-problems>.
- [5] Web Application Performance: 7 Common Problems. URL: <https://stackify.com/web-application-problems/>.
- [6] Архітектура JS Backend: підводні каміння, принципи роботи, лайфхаки. URL: <https://dou.ua/forums/topic/33590/>.
- [7] Comparison of Node.js Frameworks. URL: <https://inventorsoft.co/blog/top-14-node-js-frameworks-comparisson>.
- [8] Node.js Frameworks. URL: <https://www.geeksforgeeks.org/node-js-frameworks/>.
- [9] Оптимізація веб-сторінок та додатків. URL: <https://codeguida.com/post/189>.
- [10] 18 Tips for Website Performance Optimization. URL: <https://www.keycdn.com/blog/website-performance-optimization>.
- [11] 6 Ways to speed up your Vue.js application. URL: <https://betterprogramming.pub/6-ways-to-speed-up-your-vue-js-application-2673a6f1cde4>.
- [12] *Jeremy L. Wagner*, Web Performance in Action // 2016 – 376 p.
- [13] 9 tricks to eliminate render blocking resources. URL: <https://blog.logrocket.com/9-tricks-eliminate-render-blocking-resources/#load-custom-fonts-locally>.
- [14] Mastering browser cache. URL: <https://vueschool.io/articles/vuejs-tutorials/vue-js-performance-mastering-cache/>.
- [15] KeepAlive, Rendering Mechanism, Performance Vue. URL: <https://vuejs.org/guide/introduction.html>.

The article was delivered to editorial staff on the 05.09.2022



І.В. Груздо¹, І.В. Кириченко², Р.О. Тарасов³

¹ХНУРЕ, м. Харків, Україна, irina.gruzdo@nure.ua. ORCID iD: 0000-0002-4399-2367

²ХНУРЕ, м. Харків, Україна, iryna.kyrycheno@nure.ua. ORCID iD: 0000-0002-7686-6439

³ХНУРЕ, м. Харків, Україна, roman.tarasov1@nure.ua

АНАЛІЗ ЗАЛЕЖНОСТІ ВИБОРУ ОСВІТНІХ АЛЬТЕРНАТИВНИХ ДИСЦИПЛІН ВІД ТЕНДЕНЦІЙ ІТ ГАЛУЗІ

Була проаналізована залежність вибору альтернативних дисциплін студентами на прикладі освітньої програми 121 Інженерія програмного забезпечення для різних років навчання від тенденцій ІТ галузі. Частина даних була взята з репозиторію на платформі Github, а у якості метрик — кількісні показники існуючих тенденцій у ІТ галузі. Інформація про студентів була взята для аналізу з даних про вибір альтернатив студентами 3–4 курсу з спеціальності 121 Інженерія програмного забезпечення. Дослідження проводилося в декілька етапів, спочатку дані були проаналізовані та очищено від викидів, потім вони були ще раз розглянуті за трьома показниками, в подальшому додатково були накладені ваги, відповідно до чого було проведено дослідження і отримані фінальні значення. Під час дослідження для виміру показників залежності було використано лінійний коефіцієнт Пірсона.

АЛЬТЕРНАТИВНІ ДИСЦИПЛІНИ, РЕПОЗИТОРІЇ GITHUB, ТЕНДЕНЦІЇ ІТ ГАЛУЗІ, ЛІНІЙНИЙ КОЕФІЦІЄНТ ПІРСОНА, КОРЕЛЯЦІЯ

I.V. Gruzdo, I.V. Kyrychenko, R.O. Tarasov. Analysis of the dependency of student's selection of elective courses on tendencies in IT industry. The dependency of students' selection of elective subjects for 121 Software Engineering study plan on tendencies in IT industry was analyzed during the research on the example of students from different years of study. Part of the data was gathered from Github repositories and quantitative indicators of tendencies in IT industry were used as metrics. Information for analysis regarding students was taken from the data about students' selection for 3–4 year of studies on 121 Software Engineering program. The research was conducted in multiple stages, on the first stage the data was preprocessed and cleaned from outliers, subsequently it was analyzed one more time and converted in three numbers, then the numbers were mapped with wages, with respect to which the research was conducted and final value was obtained. Pearson correlation coefficient was used for calculation of dependency indicators during the research.

ELECTIVE SUBJECTS, GITHUB REPOSITORIES, TENDENCIES IN IT INDUSTRY, PEARSON CORRELATION COEFFICIENT, CORRELATION

Вступ

Альтернативні дисципліни стали невід'ємною частиною навчального процесу в університеті, та є складовою усіх навчальних планів, бо саме вони допомагають формувати різні компетенції та скіли у студентів спеціалістів у різних областях, не тільки за основним напрямком, що в свою чергу формує спеціаліста як різнобічну людину яка може вирішувати практичні задачі в рамках цих областей. Такі дисципліни існують на кожному році навчання, окрім першого і дуже часто вони пов'язані між собою та продовжують одна одну для того, щоб дати більш глибокі знання за певним напрямом, які в свою чергу роблять зі студентів конкурентоспроможних спеціалістів та візіонерів-аналітиків.

Студенту в самотужки інколи дуже важко сформулювати перелік дисциплін, щоб задовольнити всі необхідні потреби майбутніх роботодавців відповідно до потреб сьогодення та відносно існуючих вимог до необхідного спеціаліста [1]. Також, треба враховувати можливі ситуації в освітньому процесі, коли жоден студент не обере дисципліну або занадто багато студентів запишеться до якогось викладача і тоді треба знову перераховувати навчальне навантаження. Як наслідок може бути ситуація, коли студент обрав дисципліну, через те що веде її якийсь певний викладач

(який йому імпонує і в якого він найкраще засвоює матеріал), але через непомірне навантаження у цього викладача, дисципліну передають іншому викладачу і студент вже не бажає вивчати цю дисципліну під керівництвом іншого. У цьому випадку також порушується навантаження на семестр і треба швидко реагувати на всі можливі зміни. З урахуванням вище зазначеного можна стверджувати, що кращим вирішенням є забезпечення наявності рада альтернативних дисциплін за вибором студента, серед яких вони зможуть обирати необхідні за своїм вподобанням (вплив ряду факторів) для формування необхідних саме їм компетенцій та скілів [2].

Тому постає ряд питань: як визначити які дисципліни є необхідними та цікавими для студентів, а які ні? Як ці вподобання корелюються з вимогами до формування навчальних планів? Які метрики використовувати для визначення актуальності тієї чи іншої дисципліни? Чи вистачить компетенції з вибіркової дисципліни, для отримання хоча б посади Trainee-ра або Junior-а за відповідною вакансією? Одним із факторів, який впливає на вибір студента 121 спеціальності є загальноосвітні тенденції та нові напрямки ІТ галузі. Якщо якась технологія стає популярною, вона автоматично зацікавлює студентів та з'являється бажання вивчити її, та навпаки, якщо

якась технологія не є широко затребуваною, тож ніхто не забажає її вчити, бо вона не має свого місця на ринку та її не можна застосувати на ринку повсемірно. Дуже добрим прикладом тут буде Flash Player, підтримка якого була закінчена 31 грудня 2020 року, тож зараз вже не має ніякого сенсу викладати цю дисципліну для студентів, окрім історичної довідки, що така технологія існувала.

Метою цієї статті є розробка та обґрунтування теоретичних та практичних рекомендацій, та існуючих залежностей щодо формування вибору освітніх альтернативних дисциплін в ІТ галузі.

Об'єктом даного дослідження є процес формування вибіркового дисциплін для студентів спеціальності 121 Інженерія програмного забезпечення, а предметом дослідження – аналіз залежності вибору студентів вибіркового дисциплін від існуючих ІТ тенденцій.

Завдання статті – визначити чи є залежність вибору студентів вибіркового дисциплін від існуючих ІТ тенденцій та, якщо є, наскільки вона сильна.

1. Формулювання гіпотез

В роботі була проаналізована залежність вибору альтернативних дисциплін студентами на прикладі освітньої програми 121 Інженерія програмного забезпечення для різних років навчання від тенденцій ІТ галузі. Частина даних була взята з Репозиторію на платформі Github, а у якості метрик обрані кількісні показники, тому що вони надають змогу виконати оцінку у якості числових показників. Були розглянуті існуючі тенденції у ІТ галузі. Інформація про студентів була взята для аналізу з даних про вибір альтернатив студентами 3-4 курсу з спеціальності 121 Інженерія програмного забезпечення. Дослідження проводилося в декілька етапів.

На початку були сформульовані гіпотези, які відображають об'єкт нашого дослідження. У ході аналізу предметної області були виділені наступні гіпотези:

H_0 – відсутність кореляційного зв'язку між академічною зацікавленістю студентів та популярності вибору альтернативних дисциплін (коефіцієнт кореляції = 0);

H_1 – кореляційний зв'язок присутній (коефіцієнт кореляції $\neq 0$).

Ці гіпотези призначені перевірити чи є залежність вибору альтернативних дисциплін в університеті від інтересів студентів та загальних трендів світу ІТ та, якщо є, то наскільки сильна ця кореляція.

Збір, аналіз і визначення зв'язків над елементами датасету є окремою масштабною задачею, тому що необхідно визначити, що саме впливає на популярність альтернативних дисциплін за вибором студента.

Під час аналізу були враховані наступні чинники: популярність дисципліни на окремих роках навчання, популярність технологій за опитуванням спеціалістів в ІТ (дані з репозиторіїв з відповідними технологіями на платформі Github).

Сутність перевірки полягає у тому, щоб отримати лінійний коефіцієнт, який може бути розподілений від 0 до 1. Значення до інтервалу, якому воно належить, буде казати наскільки сильна чи слабка кореляція спостерігається між цими величинами. Можливі значення можна побачити нижче у табл. 1.

Таблиця 1

Відповідність значення лінійного коефіцієнту кореляції до висновку

Значення	Інтерпретація
від 0 до 0.3	дуже слабка
від 0.3 до 0.5	слабка
від 0.5 до 0.7	середня
від 0.7 до 0.9	сильна
від 0.9 до 1	дуже сильна

Для того, щоб прийняти нульову гіпотезу необхідно мати слабку кореляцію і вище.

В рамках роботи над поставленою задачею не було виявлено доступних датасетів про вибір альтернатив студентами різних Вищих. Це підтверджує актуальність даного дослідження і створює певні складнощі для досягнення мети роботи.

2. Опис основних джерел та каналів даних

Слід відмітити, що для проведення дослідження необхідні специфічні дані, які майже всі не знаходяться у відкритому доступі, та потребують предоброби для подальшого їх використання.

Зі сторони університету потрібні дані про відвідуваність вибіркового дисциплін у відповідні роки, роки введення цих дисциплін, а також кількість записавшихся на ці дисципліни студентів, та тих хто потім відмовився вивчати її. Авжеж, зрозуміло, що всі ці фактори залежать не тільки від зацікавленості у предметі, а й від самого викладача, наприклад наскільки саме він як людина імponує студенту (суб'єктивний фактор кожного студента), та як наслідок чи зможе студент найкраще сприйняти та засвоїти наданий викладачем матеріал. Відповідно до цього можна стверджувати, що є багато інших різних факторів як професійних так і психологічних, які впливають на вибір студента.

Тому важливо виділити головні чинники, серед них головні з боку університету при формуванні списку вибіркового предметів [3-7]:

– це дотримання головних вимог стандарту вищої освіти (обсяг кредитів ЄКТС; перелік компетентностей випускника; вимоги професійних стандартів; форми атестації здобувачів вищої освіти та інш.), законів України «Про освіту»; «Про вищу освіту», «Про наукову та науково-технічну діяльність»;

– виконання умови, що мінімум 50% обсягу освітньої програми має бути спрямовано на забезпечення загальних та спеціальних (фахових) компетентностей за спеціальністю, визначених стандартом вищої освіти;

- дотримання умов, що всі дисципліни які викладаються в рамках навчального плану повинні поділяються на цикли загальної та професійної підготовки;
 - у межах відповідної освітньої програми та робочого навчального плану передбачена можливість для обрання студентом дисциплін за вибором, що становить не менше як 25% загальної кількості кредитів ЄКТС, з урахуванням власних потреб, інтересів, уподобань щодо майбутньої професійної діяльності або розширення особистісного світогляду;
 - загальна кількість навчальних дисциплін, запланованих до вивчення, регламентується трудомісткістю необхідних виконаних навчальних робіт, що становить 60 кредитів ЄКТС на навчальний рік з розподілом за семестрами та не повинна перевищувати 8 дисциплін на кожний навчальний семестр;
 - випускові кафедри в межах освітньої програми та навчального плану для підготовки бакалаврів повинні формувати цикл професійної підготовки, який містить перелік вибіркового компонентів освітньої програми які, як правило, стосуються підготовки здобувачів за освітньою програмою та відповідають запитам підприємств-роботодавців;
 - результати освітнього процесу повинні відповідати цілям освітньої програми, очікуванням і потребам студентів та суспільства, при цьому забезпечувати можливість працевлаштування за вимогами підприємств-роботодавців;
 - та інші.
- Як бачимо, з цього перед колективами випускових кафедр постає складне питання розробки наукових планів із підготовки бакалавра, магістра, доктора філософії, які мають бути орієнтиром для студентів при виборі напрямів навчання, а також допомогти в оволодінні загальними та фаховими компетенціями, що відповідають запитам підприємств-роботодавців;
- Головні чинники, що впливають на вибір дисципліни (вибіркової) з боку студента:
- шкільний досвід та навчальні результати за перший рік навчання в університеті (для студентів другого року навчання);
 - особистий та інтелектуальний розвиток студента;
 - мотиваційні чинники, що до можливості отримати хорошу роботу в майбутньому;
 - здобуття професійних знань та навичок за відповідним напрямом;
 - зацікавленість предметом через те що він є найбільш популярним в галузі;
 - зміст, тривалість, наповнення, система оцінювання за дисципліною;
 - кваліфікація викладача та на скільки студенту добре працювати (сприймати матеріал) від певного лектора;
 - затребуваність тієї чи іншої технології/ компетенції/скілів на ринку праці;
 - як збільшиться майбутній зарібок, якщо

студент буде володіти та розбиратися в напрямі за відповідним предметом;

- чи суміщає студент навчання з роботою, наскільки йому знадобиться в майбутньому цей предмет за його теперішньою роботою;

- попередні вибори дисциплін студентом та бажання продовжувати навчання за відповідним напрямком.

Тобто можна стверджувати, що з боку студента є одним із головних факторів, це те, що потрібно враховувати тенденції ІТ індустрії для формування освітнього плану. Щоб це підтвердити необхідні дані з кафедри чи університету які б враховували всі ці головні чинники. По перше це дані про вибір дисципліни студентом, анінні опитування студентів а також опис кожної з дисциплін за вибором.

З іншої сторони дані будуть збиратися с платформи Github, на якій можна відстежувати популярність тих чи інших технологій за окремі проміжки часу. Ця інформація може бути отримана з Github API [8].

Також обидва канали потребують попередньої обробки даних, таких як нормалізацію даних, та виведення занадто екстремальних значень з вибірки для того, щоб не зіпсувати увесь набір даних. А саме дані з бази даних університету потребує виділення лише тих предметів, котрі були вже більше 3 років у виборі студентів, задля того, щоб видалити ймовірно погрішності через суб'єктивізм до кваліфікації викладачів та можливе небажання студентів йти на зовсім нову дисципліну. Зі сторони Github потрібно обчислити відповідь з API, щоб отримати тільки кількість зірок (stars), форків (fork) та спостерегань (watch).

Такі дані потрібно обраховувати за 4-5 років, для того, щоб отримати більш повну статистику за новою дисципліною.

3. Характеристика вибірки

Для того, щоб визначити достатній об'єм вибірки, можна скористатися наступною формулою (обчислення потужності вибірки):

$$n = \frac{t^2 p(100 - p)}{\Delta^2},$$

де: t – довірчий рівень, статистична величина, значення якої для досліджень в соціальній сфері прийнято 1,96 (при 95% точності статистичного висновку). Довірчий рівень встановлює сам дослідник відповідно до своїх вимог до надійності отриманих результатів. Найчастіше застосовуються довірчі рівні, рівні 0,95 або 0,99; p – % об'єктів, у яких ймовірно проявляється ознака, важлива для проведеного дослідження; Δ – допустима помилка в %, задається довільно при плануванні дослідження.

Для нашого аналізу було обрано наступні значення: $t = 0.95$, це найнижча планка серед найчастіше використовуваних; $\Delta = 12 \%$, бо наша задача виявити таку залежність і точність тут не так важлива, як сам факт наявності залежності; $p = 80 \%$, бо не

всі технології є open source та мають репозиторії на Github.

Отримали наступне:

$$n = \frac{0,95^2 80(100 - 80)}{12^2} = \frac{1444}{144} = 10.$$

Іншим джерелом даних став Github API, який має архітектуру REST та може бути використаний будь-якою мовою програмування. На рис. 1 наведений приклад доступу до даних через термінал.

4. Угрупування даних вибірки

У якості кількісної характеристики було обрано кількість репозиторіїв, “лайки” котрих перевищує $2k$ (під лайками мається на увазі stars).

Мінімальне значення вибірки — 1 репозиторій, максимальне значення — 20.

Угрупування було виконано з відносно рівним інтервалом за формулою Стерджесса, адже величина інтервалу залежить від розмаху варіювання ознаки і чисельності досліджуваної сукупності. Формула Стерджесса для обчислення інтервалу групування наведена нижче

$$i = \frac{x_{\max} - x_{\min}}{k} = \frac{x_{\max} - x_{\min}}{1 + 3,322 * \lg N}$$

Кількість елементів була обрахована у попередній формулі та вона дорівнює 10.

Тож зараз можемо обрахувати i та k .

$$i = \frac{20 - 1}{1 + 3,322 * \lg 10} = 4,5,$$

$$k = 1 + 3,322 * \lg 10 = 4,222 \Rightarrow 4.$$

Тобто маємо 4 групи та інтервал між ними приблизно 4.5.

Розподілені дані за групами можна побачити у табл. 2.

Таблиця 2

№ групи	Кількість репозиторіїв
1	1 — 5.5
2	5.5 — 10
3	10 — 14.5
4	14.5 — 20

5. Підрахунок коефіцієнту кореляції

Далі необхідно обрати репозиторії для окремих тем, які викладаються в університеті та для яких маємо статистику відвідувань та успішності. Для того, щоб знайти репозиторії за якоюсь темою. Наприклад, на рис. 2 можемо побачити як виконується пошук за темою NoSQL з використанням Github UI.

Все це робиться з використанням Github API. В процесі аналізу потрібно звернути увагу на кількість зірок та обирати ті репозиторії, які мають більше $2k$ зірок та $1k$ форків, тобто це означає, що на дану секунду цей репозиторій є популярним, та в нього багато розробників бажає законтрибуїтисся.

Тож фрагмент результату зображено у табл. 3

Таблиця 3

Результати обробки репозиторіїв	
Тема	Кількість репозиторіїв (Y)
.NET	18
NoSQL	16
Java	13
IoT	3
Computer Vision	10
Python	17
Networking	6
Data analysis	20
Test automation	8
Unity	9

```
rtarasov@su-macbookpro-b60d:~$ curl -H "Accept: application/json" https://api.github.com/repos/python/cpython
{
  "id": 81598961,
  "node_id": "MDEwO1JlcG9zaXRvcnk4MTU5ODk2MQ==",
  "name": "cpython",
  "full_name": "python/cpython",
  "private": false,
  "owner": {
    "login": "python",
    "id": 1525981,
    "node_id": "MDEyOjk9yZ2FuaXphdGlvdjE1MjU5ODE=",
    "avatar_url": "https://avatars.githubusercontent.com/u/1525981?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/python",
    "html_url": "https://github.com/python",
    "followers_url": "https://api.github.com/users/python/followers",
    "following_url": "https://api.github.com/users/python/following{/other_user}",
    "gists_url": "https://api.github.com/users/python/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/python/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/python/subscriptions",
    "organizations_url": "https://api.github.com/users/python/orgs",
    "repos_url": "https://api.github.com/users/python/repos",
    "events_url": "https://api.github.com/users/python/events{/privacy}",
    "received_events_url": "https://api.github.com/users/python/received_events",
    "type": "Organization",
    "site_admin": false
  },
}
```

Рис. 1. Приклад запиту до Github API

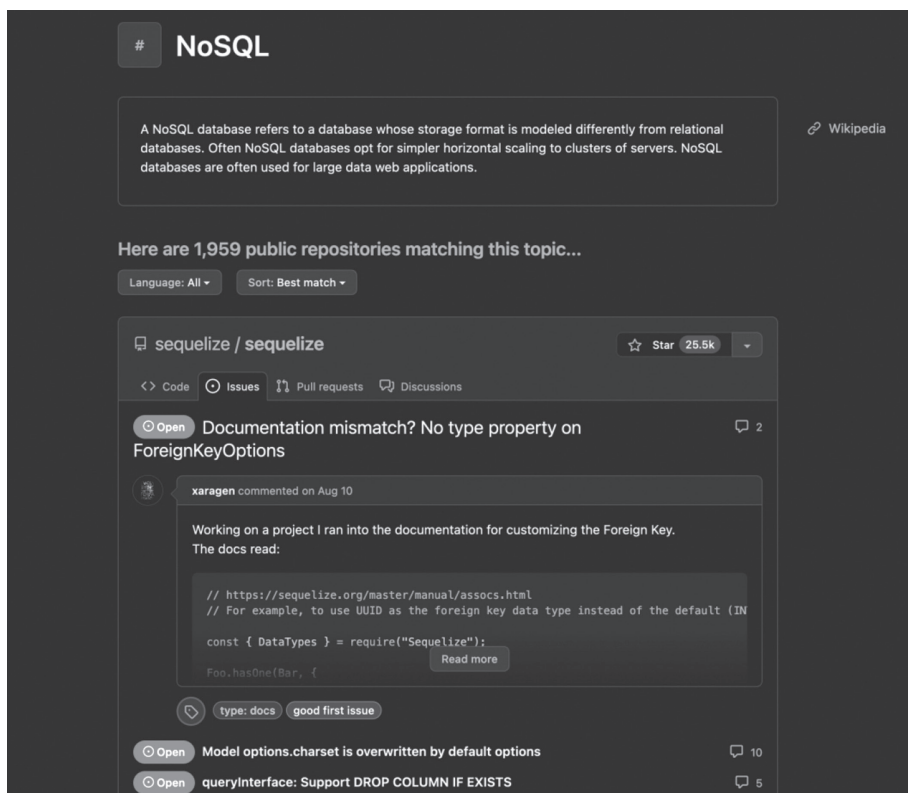


Рис. 2. Пошук за темою NoSQL

Відповідно до проведеного аналізу можна стверджувати, що у табл. 3 зібрані всі найпопулярніші теми, які викладаються в університеті на альтернативних дисциплінах спеціальності 121 Інженерія програмного забезпечення.

Далі було розраховано коефіцієнти альтернативних дисциплін з урахуванням необхідної ваги, заформулою:

$$5 * a + 3 * b + 2 * c,$$

де: a — середній процент відвідування альтернативи протягом семестру; b — процент заповненості альтернативи; c — середній бал студентів.

У табл. 4 зображено розрахунки по усім альтернативним дисциплінам відносно переліку тем з табл. 3.

Таблиця 4

Результати обробки репозиторіїв

Тема	Коефіцієнт альтернативної дисципліни (X)
.NET	$5 * 0.5 + 3 * 1 + 2 * 0.88 = 7.26$
NoSQL	$5 * 0.75 + 3 * 1 + 2 * 0.85 = 8.45$
Java	$5 * 0.6 + 3 * 0.8 + 2 * 0.8 = 7$
IoT	$5 * 0.1 + 3 * 0.1 + 2 * 0.7 = 2.2$
Computer Vision	$5 * 0.7 + 3 * 1 + 2 * 0.9 = 8.3$
Python	$5 * 0.9 + 3 * 1 + 2 * 0.8 = 9.1$
Networking	$5 * 0.6 + 3 * 1 + 2 * 0.9 = 7.8$
Data analysis	$5 * 0.5 + 3 * 1 + 2 * 0.93 = 7.36$
Test automation	$5 * 0.6 + 3 * 0.9 + 2 * 0.85 = 7.4$
Unity	$5 * 0.85 + 3 * 1 + 2 * 0.8 = 8.65$

Далі було розраховано:

- середнє значення для Y : $120 / 10 = 12$;
- середнє значення для X : $73.52 / 10 = 7.352$;
- середнє значення для $X * Y$: $931.75 / 10 = 93.175$.

Наступним кроком було виконано пошук дисперсії

$$D(Y) = 360 / 9 = 60$$

$$D(X) = 33.72 / 9 = 3.75$$

Знайдене середньоквадратичне відхилення:

$$\sigma(x) = \sqrt{60} = 7.75,$$

$$\sigma(y) = \sqrt{3.75} = 1.9.$$

Далі пораховано dX та dY (як дельта від середнього значення) та отримані дані наведено в табл. 5.

Потім розраховано коефіцієнт кореляції за наступною формулою [9-10]:

$$r_{xy} = \frac{\overline{x \cdot y} - \bar{x} \cdot \bar{y}}{\sigma(x) \sigma(y)},$$

$$r_{xy} = \frac{93,175 - 12 * 7,352}{7,75 * 1,9} = \frac{4,951}{14,725} = 0,34.$$

Якщо $|r| = \pm 1$, то кореляційний зв'язок між X і Y являє собою лінійну залежність. Кореляція є позитивною та слабкою, адже збільшення значення однієї змінної призводить до збільшення значення другої (чим більше популярних репозиторіїв, тим більше академічна зацікавленість студентів).

Рахуємо значущість коефіцієнту кореляції:

$$t = r * \frac{\sqrt{n-2}}{\sqrt{1-r}} = 0,33 * 3 = 0,99.$$

Таблиця 5

Всі параметри для підрахунку коефіцієнта r

Тема	Y	X	X * Y	dX	dY
.NET	18	7.26	130.68	-6	0.09
NoSQL	16	8.45	135.2	-4	-1.098
Java	13	7	91	-1	0.352
IoT	3	2.2	6.6	9	5.152
Computer Vision	10	8.3	83	2	-0.948
Python	17	9.1	154.7	-5	-1.748
Networking	6	7.8	46.8	6	-0.448
Data analysis	20	7.36	147.2	-8	-0.008
Test automation	8	7.4	59.2	4	-0.048
Unity	9	8.65	77.85	3	-1.298

Для оцінки значущості коефіцієнту кореляції було використано критерій Стьюдента. По таблиці Стьюдента знаходимо:

$$T(n-2; t/2) = T(8; 0,5) = 7,34$$

Так як $T_{\text{набл}} = 0,99 < T_{\text{кр}} = 7,35$, тобто так як значення менше критичного, то приймається нульову гіпотезу, про відсутність кореляційного зв'язку між академічною зацікавленістю студентів та популярності вибору альтернативних дисциплін. Все це свідчить про те що при виборі дисциплін, студенти керуються різними причинами, які в свою чергу інколи суперечать один одному. Тому необхідно, щоб більш компетентні спеціалісти приймали більш збільшене рішення яке буде враховувати не лише потреби ринку ІТ, а й враховувало існуючі нормативні акти відносно формування навчальних планів.

Висновки

В межах даної статті було розглянуто головні чинники які впливають на формування та вибір дисциплін за навчальним планом, як з боку університету так і з боку студента. Відповідно до цього були виділені критерії за якими проводилося дослідження метою якого було визначення залежності вибору альтернативних дисциплін студентами від тенденцій ІТ галузі. Були визначені джерела даних, на основі яких проводилося дослідження. В ході дослідження було визначено, що коефіцієнт кореляції підпадає під слабкий рівень, тож можемо зробити висновок, що нульова гіпотеза приймається та існує прямо пропорційна залежність між тенденціями ІТ галузі та вибором альтернативних дисциплін. Цей зв'язок є слабким, але також усі ці дані були тільки за три роки, якщо доробити вибірку, за 5 років можна отримати більш точні значення. Тобто можна зробити загальний висновок, що студенту в самотужки дуже важко сформулювати перелік дисциплін, щоб задовольнити всі необхідні потреби майбутніх роботодавців відповідно до потреб сьогодення та відносно існуючих вимог до необхідного спеціаліста? Тому обов'язково необхідно надавати анотації до дисциплін та хоча б інколи проводити оцінку якості кваліфікації ІТ-спеціалістів [11].

Список літератури:

- [1] Шрамченко Б. Л., Щербань В. Ю. Планування навчального процесу дисциплін за вибором студента // Вісник Київського національного університету технологій та дизайну. – 2010. С. 245-246.
- [2] Журан О.А., Філатова Т.В., Чернишов О.О. Модель формування сучасних компетенцій ІТ-фахівців. Інформатика та математичні методи в моделюванні. 2019. № 9(3). С. 195–202. DOI: <https://doi.org/10.15276/imms.v9.no3.195>
- [3] Положення про організацію освітнього процесу (далі – Положення) в Харківському національному університеті радіоелектроніки Наказ від 27.11.2020 №400 [Електронний ресурс] – Режим доступу до ресурсу: https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-pro-organizaciju-osvitnogo-procesu-v-hnure.pdf. Дата звернення 17.12.2021.
- [4] Закон України «Про освіту» [Електронний ресурс] // Верховна Рада України. – Режим доступу <https://zakon.rada.gov.ua/laws/show/2145-19#Text>. Дата звернення 17.12.2021.
- [5] Закон України «Про вищу освіту» [Електронний ресурс] // Верховна Рада України. – Режим доступу <https://zakon.rada.gov.ua/laws/show/1556-18#Text>. Дата звернення 17.12.2021.
- [6] Закон України «Про наукову і науково-технічну діяльність» [Електронний ресурс] // Верховна Рада України. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/848-19#Text>. Дата звернення 17.12.2021.
- [7] Міністерство освіти і науки України. Професійний стандарт. Фахівець з інформаційних систем [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://mon.gov.ua/ua/osvita/visha-osvita/suchasna-it-osvita-vukrayini/profesiyni-standarti>. Дата звернення 17.12.2021.
- [8] Github REST API [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.github.com/en/rest>. Дата звернення 17.12.2021.
- [9] Edwards, A. L. “The Correlation Coefficient.” Ch. 4 in An Introduction to Linear Regression and Correlation. San Francisco, CA: W. H. Freeman, pp. 33-46, 1976.
- [10] Gonnick, L. and Smith, W. “Regression.” Ch. 11 in The Cartoon Guide to Statistics. New York: Harper Perennial, pp. 187-210, 1993.
- [11] Литвинов В. В., Трунова О. В., Войцеховська М. М. МАТЕМАТИЧНА МОДЕЛЬ ОЦІНКИ УЗАГАЛЬНЕНОГО КРИТЕРІЮ ЯКОСТІ ПІДВИЩЕННЯ КВАЛІФІКАЦІЇ ІТ-СПЕЦІАЛІСТІВ // МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ Черкаський національний університет імені Богдана Хмельницького Черкаський інститут банківської справи Чорноморський державний університет імені Петра Могили. – С. 225-228.

Надійшла до редколегії 14.10.2022



Ilona Revenchuk¹, Vladyslav Valuiskyi², Mykhailo Temnokhud³

¹ Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
ilona.revenchuk@nure.ua, ORCID: 0000-0002-5188-9538

² Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
vladyslav.valuiskyi@nure.ua

³ Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
mykhailo.temnokhud@nure.ua

WIRELESS METHODS FOR THE LOCAL POSITIONING WITHOUT ACCESS TO THE INTERNET AND GPS SERVICE

Local positioning in buildings without access to the Internet is a crucial requirement in many scenarios, such as large corporate offices, hospitals, universities, and shopping malls. However, this need for local positioning also presents certain challenges and problems that must be addressed. One of the most significant challenges of local positioning in buildings without internet access is the lack of reliable and up-to-date indoor map data. Without accurate map data, it can be difficult to navigate through complex indoor environments or locate specific rooms or facilities within a building. Another challenge is the absence of GPS signals in indoor environments, which makes it challenging to obtain accurate location information. This is especially problematic when attempting to track the movement of people or objects within a building. Additionally, the lack of internet access can make it difficult to implement location-based services that rely on data transfer between devices. This includes applications like indoor navigation, asset tracking, and location-based marketing, which all require internet connectivity to function correctly. Despite these challenges, there is a growing need for local positioning in buildings without internet access. This is especially true for large organizations and institutions that rely on efficient and effective operations within their indoor environments. Some technologies can provide accurate location information and help to mitigate some of the challenges associated with local positioning in buildings without internet access. They allow to obtain the accurate location information without relying on GPS signals or internet connectivity. The need for local positioning in buildings without internet access is becoming increasingly important, especially for large organizations and institutions that require efficient and effective operations within their indoor environments.

LOCAL POSITIONING METHODS, WIRELES TECHNOLOGIES, WI-FI, BLE, IPS

І.А. Ревенчук, В.С. Валуйський, М.І. Темнохуд. Бездротові методи локального позиціонування без доступу до Інтернету та служби GPS. Локальне позиціонування в будівлях без доступу до Інтернету є важливою вимогою в багатьох сценаріях, наприклад у великих корпоративних офісах, лікарнях, університетах і торгових центрах. Однак ця потреба в локальному позиціонуванні також створює певні виклики та проблеми, які необхідно вирішити. Однією з найбільш важливих проблем локального позиціонування в будівлях без доступу до Інтернету є відсутність надійних і актуальних даних карт приміщень. Без точних картографічних даних може бути складно орієнтуватися в складних приміщеннях або визначати місцезнаходження окремих приміщень чи об'єктів у будівлі. Іншою проблемою є відсутність сигналів GPS усередині приміщень, що ускладнює отримання точної інформації про місцезнаходження. Це особливо проблематично під час спроби відстежити рух людей або об'єктів усередині будівлі. Крім того, відсутність доступу до Інтернету може ускладнити впровадження служб на основі визначення місцезнаходження, які покладаються на передачу даних між пристроями. Сюди входять такі додатки, як навігація в приміщенні, відстеження ресурсів і маркетинг на основі місцезнаходження, для належної роботи яких потрібне підключення до Інтернету. Незважаючи на ці проблеми, зростає потреба в локальному позиціонуванні в будівлях без доступу до Інтернету. Це особливо вірно для великих організацій та установ, які покладаються на ефективну та результативну роботу всередині своїх приміщень. Деякі технології можуть надати точну інформацію про місцезнаходження та допомогти пом'якшити деякі проблеми, пов'язані з локальним позиціонуванням у будівлях без доступу до Інтернету. Вони дозволяють отримати точну інформацію про місцезнаходження, не покладаючись на сигнали GPS або підключення до Інтернету. Потреба в локальному позиціонуванні в будівлях без доступу до Інтернету стає все більш важливою, особливо для великих організацій та установ, які потребують ефективної та результативної роботи всередині своїх приміщень.

МЕТОДИ ЛОКАЛЬНОГО ПОЗИЦІОНУВАННЯ, БЕЗДРОТОВІ ТЕХНОЛОГІЇ, WI-FI, BLE, IPS

Introduction

Wireless local positioning without access to the internet and GPS service has become an essential requirement for many industries such as logistics, healthcare, manufacturing, and retail. With the increasing demand for location-based services, there is a growing need for reliable and accurate wireless local positioning methods that can work even in environments where GPS signals and internet connectivity are not available.

Traditional GPS-based systems cannot provide accurate location information in indoor environments, such as shopping malls, hospitals, and airports, where signals are often blocked or degraded by walls and other obstacles. In addition, many remote areas or developing countries still lack reliable internet connectivity, making it challenging to provide location-based services that rely on internet access.

To overcome these challenges, new wireless methods have been developed for local positioning without access

to the internet and GPS service. These methods use wireless technologies such as Bluetooth Low Energy (BLE) beacons, Wi-Fi-based positioning, and indoor positioning systems (IPS) to provide accurate location information. The use of these technologies can be highly beneficial for industries such as logistics, where accurate location information can improve efficiency and reduce costs.

1. Wi-Fi network positioning general description

Nowadays, Wi-Fi networks are so common that with their help you can easily create a network that will consist of a large number of devices that are not tied to a specific location and can change their position in space. At the moment, almost any corporate client has Wi-Fi coverage, so the infrastructure of such a wireless network can be used to perform a number of specific tasks. One of them can be local positioning in the organization. The accuracy of such systems directly depends on the density of access points that are tied to specific points on the plan of a building, structure or a certain area.

Indoor Wi-Fi solutions use existing Wi-Fi access points or special Wi-Fi enabled sensors to detect and locate Wi-Fi transmitting devices such as smartphones and indoor tracking tags[1].

Location data collected by sensors or access points or sent from access points to client devices is fed into various location programs and converted into statistical data, which in the future can provide numerous cases of use of user location data.

Wi-Fi-based positioning systems can use different methods to determine the location of devices. Most of them rely on methods that are based on the Received Signal Strength Indicator (RSSI). However, some apps may use more advanced Wi-Fi positioning methods[2].

1.1. Wi-Fi positioning using access points

Wi-Fi positioning using access points directly depends on the existing Wi-Fi infrastructure installed indoors to determine the location of the desired devices[2].

This technology allows large organizations to use their existing infrastructure to run their location-based applications without the need to install new hardware.

Access points in a building can detect data transmissions from surrounding Wi-Fi devices, both on and off the local network. This data about the user's location is then sent to the server and used to calculate the position of the device in the space of the building.

1.2. Wi-Fi positioning using sensors

Sensor-assisted Wi-Fi positioning primarily uses Wi-Fi-enabled sensors that are deployed at fixed positions within the interior of a building.

These sensors can passively detect and locate data transmissions from smartphones, asset tracking tags, beacons, personnel badges, portable devices and other Wi-Fi devices.

The location data collected by the sensor is then sent to the server and received by the central Indoor Positioning System (IPS) or Real-Time Location System (RTLS). The location engine analyzes the data to determine the location of the device transmitting the data[3]. These coordinates can be used to visualize the location of a device or object on a spatial map of a particular space or be used for other purposes depending on the specific application that needs to recognize the location.

2. Wi-Fi positioning methods

The most common methods of Wi-Fi positioning determine the location of an object using an indicator called the Received Signal Strength Indicator (RSSI), mainly for device multilateration or fingerprint calculation. These approaches, which were based on signal strength, are very simple to implement and cost-effective, but cannot provide high accuracy because the signal strength can be affected by the environment.

RSSI solutions are also susceptible to errors that can be caused by the movement of objects in the environment, such as people.

Adding other less common and more advanced methods can lead to more accurate Wi-Fi positioning results. These include Angle of Arrival (AoA) and Time of Flight (ToF)

2.1. RSSI multilateration

In RSSI-based applications, multiple existing Wi-Fi access points or fixed-position Wi-Fi-enabled sensors will detect Wi-Fi devices that are transmitting data and the received signal strength from the device.

This location data collected by access points or sensors will be sent to a central Indoor Positioning System (IPS) or Real-Time Location System (RTLS)[4].

The RSSI multilateration scheme is shown in fig. 1.

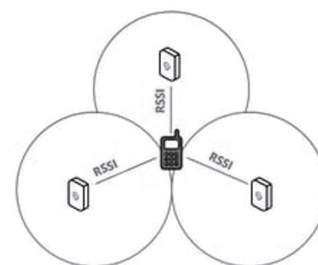


Fig. 1. RSSI scheme

The exact location engine analyzes the data and uses multilateration algorithms to estimate the location of the devices transmitting the data. In addition, you can use the signal strength of the nearest access points relative to the wireless device to determine the location of the device.

Using an RSSI-based method with multilateration is the simplest and cheapest option for Wi-Fi positioning. However, it cannot provide a high degree of positional accuracy because it is subject to signal attenuation, signal absorption, signal reflection and various interferences.

2.2. RSSI fingerprinting

Fingerprinting is also an RSSI-based method. Wi-Fi fingerprint positioning primarily involves the use of a database that will record the location and signal strength of surrounding Wi-Fi access points, as well as the Wi-Fi coordinates of a device such as a smartphone or tracking tag in an inactive phase[5].

Creating a fingerprint database requires a long and time-consuming calibration process that may need to be repeated multiple times. During active device tracking, the RSSI value is compared to these footprints in the database to estimate the location of the pre-trained device.

The RSSI fingerprinting scheme is shown in fig. 2.

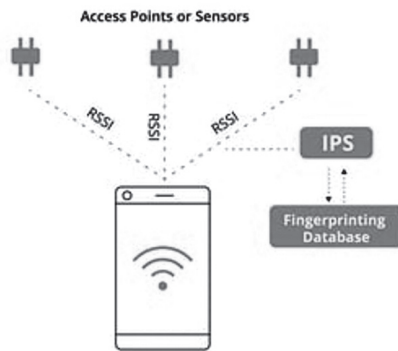


Fig. 2. RSSI fingerprinting scheme

Similar to signal strength and multilateration positioning, fingerprinting does not provide high positioning accuracy unless the system is constantly calibrated to environmental changes. This is a low-cost Wi-Fi positioning method, but it requires constant updating of the trained patterns in the database.

Fingerprinting approaches are also affected by signal attenuation (the biggest influence), signal absorption, signal reflection, and random interference in the signal path.

2.3. Time of Flight (ToF) method

Time of Flight – it is a highly accurate positioning method used by precision technologies such as UWB. This advanced technique can accurately measure the distance between Wi-Fi devices by calculating the time it takes for a signal to travel between devices.

ToF can be used to determine the exact location of a Wi-Fi device using multiple sensors or access points. This requires an integrated deployment of access points or sensors to detect Wi-Fi devices such as a smartphone or tracking tag.

To work properly, sensors or Wi-Fi access points must be precisely synchronized to the same master clock. The device’s Wi-Fi signals will be received by access points or sensors in range and time-stamped.

The scheme of ToF method is shown in fig. 3.

All time-stamped data is then sent to a central ISP or RTLS. The location system will analyze the data from each anchor and the difference in arrival time at each anchor, and then use multilateration to accurately calculate the coordinates of the mark.

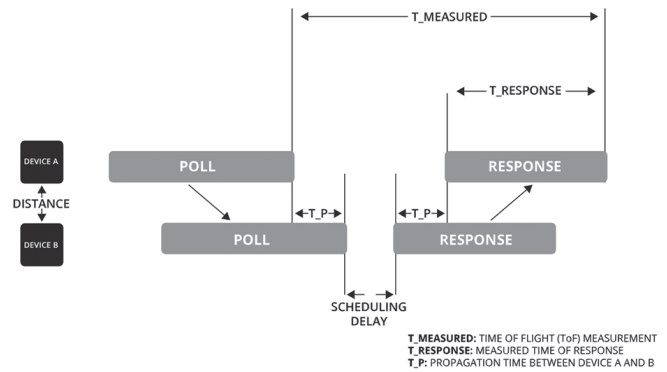


Fig. 3. Time of Flight scheme

Although ToF provides more accurate Wi-Fi positioning, the deployment of this approach involves a higher level of complexity and may not be cost-effective in scenarios where high accuracy is not required. Wi-Fi Round-Trip-Time (Wi-Fi RTT) is a new method of locating Wi-Fi devices that uses ToF. Specified in the IEEE 802.11-2016 standard, Wi-Fi RTT allows devices to measure the distance between devices using the signal transit time between devices.

This calculation is based on the path traveled by the signal and is determined using the speed of the electromagnetic wave and the speed of light. This can be used for ranging between two devices or for indoor positioning with multiple access points or multi-lateration sensors.

2.4. Angle-of-Arrival (AoA) method

The angle-of-arrival method is an advanced method that can provide Wi-Fi positioning with increased accuracy compared to more traditional methods such as fingerprinting and RSSI.

This is made possible thanks to the Wi-Fi interface Multiple Input Multiple Output (MIMO). To be able to determine direction, a mobile device such as a tag or beacon with a single antenna transmits data to a fixed Wi-Fi sensor with a multi-antenna system.

The phase shift of multiple antennas as a result of signal reception is measured and calculated to determine the angle of the transmitting mobile device and create a zone of certainty of the object to be found.

One advantage of the angle of arrival method is that it reduces the number of reference points required. Instead of the minimum three sensors required for any multilateration method, you only need two to determine position unambiguously.

The scheme of AoA is shown in fig. 4.

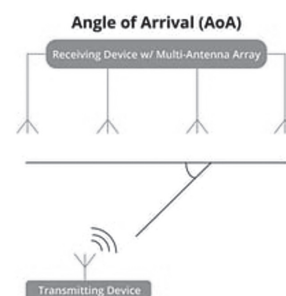


Fig. 4. Angle of Arrival scheme

Additional reference points increase the accuracy and reliability of calculated positions. Although angle-of-arrival indoor positioning is more accurate than other methods up to signal strength, solutions using this technique are just entering the market.

3. Wi-Fi positioning accuracy, range and frequency of work

So how accurate can Wi-Fi positioning be? Wi-Fi 5 and below are typically less accurate than other RF technologies such as Ultra Wideband Beacons (UWB) and Bluetooth Low Energy (BLE) and achieve Wi-Fi location accuracies typically less than 10 meters (under optimal conditions and deployment).

The current Wi-Fi 6 technology standard promises accuracy in the meter range, but since this is a new technology, this has not yet been demonstrated and tested in the field. The last update to the 802.11az standard in March 2021, also called Next Generation Positioning (NGP), promises improved Wi-Fi positioning accuracy.

Wi-Fi positioning accuracy is significantly lower when using single access points, but Wi-Fi-enabled sensors can be added to enhance the tracking of traditional access points, or even as separate sensors for more accurate positioning in the vicinity.

Different Wi-Fi positioning methods can also provide different degrees of accuracy. Traditional approaches such as RSSI multilateration and fingerprinting provide accuracies that are significantly lower than more advanced methods such as angle of arrival, time-of-flight, and Wi-Fi RTT.

The Wi-Fi positioning range may vary depending on factors such as the use of Wi-Fi access points or sensors or the nature of the indoor space. Wi-Fi operating at 2.4 GHz typically works up to 100 meters (under optimal conditions and deployment) and can be used even outside of premises where the appropriate infrastructure is in place.

However, Wi-Fi operating at 5 GHz reduces the range due to the higher frequency, operating at about 50% of the achievable range of 2.4 GHz. Also, at 5 GHz, higher signal attenuation should be expected if signals must pass through environmental obstructions such as doors, walls, or metalized windows.

Wi-Fi technology, like other RF standards, offers unique characteristics and benefits that can make it an accepted option depending on individual needs, project budget, facility, and specific location-based use cases.

The most important differences between Wi-Fi and other technologies are its ability to use existing Wi-Fi infrastructure and its flexibility for use in many applications that use location. Wi-Fi is present in devices almost everywhere, is used by many location tracking systems, and can be extended for indoor positioning in a whole range of industries and use cases.

4. Bluetooth Low Energy technology overview

Bluetooth Low Energy (BLE) is a wireless communication technology that was introduced as a part of the Bluetooth 4.0 specification. It is designed to provide low-power wireless connectivity to devices with low data rates over a short range. BLE operates on the same 2.4GHz ISM band as traditional Bluetooth, but it uses a different modulation scheme and has a lower power consumption than traditional Bluetooth[6].

This technology was developed to address the limitations of traditional Bluetooth, which required high power consumption and was not suitable for applications that required long battery life. BLE provides a low-power alternative to traditional Bluetooth, allowing devices to run on batteries for months or even years, making it an ideal solution for devices such as sensors, wearables, and IoT devices.

BLE uses a master-slave architecture in which a master device controls the communication with one or more slave devices. The master device is responsible for initiating and managing the communication, while the slave device responds to the master's requests. The communication between the master and slave devices is carried out using packets of data that are transmitted over the airwaves.

One of the key features of BLE is its ability to operate in a low-power mode, also known as sleep mode. In sleep mode, the device consumes very little power and wakes up only when there is data to be transmitted or received. This allows devices to conserve battery life and operate for long periods without the need for frequent recharging.

It is also designed to be highly secure, with features such as encryption and authentication built into the protocol. This makes it suitable for applications that require a high level of security, such as healthcare, finance, and government.

This technology is widely used for applications that require low power consumption and short-range wireless connectivity, such as wireless sensors, wearables, and IoT devices[7]. BLE enables devices to connect and communicate with each other over a range of up to 100 meters in open space. However, the range can be reduced by walls and other obstacles.

BLE is often used in combination with other wireless technologies, such as Wi-Fi and GPS, to provide more accurate location information in indoor environments or areas where GPS signals are not available. BLE beacons, for example, are small battery-powered devices that emit a signal that can be received by BLE-enabled devices, allowing them to determine their proximity to the beacon and calculate their location.

It is also widely used in healthcare applications, where it can be used to monitor patients' vital signs, track their medication usage, and provide remote patient monitoring. In addition, BLE-enabled wearables can be used to

track fitness and exercise activities, such as steps taken, distance traveled, and calories burned.

So, BLE technology offers low power consumption, short-range wireless connectivity, and a wide range of applications in various industries, including healthcare, fitness, automotive, and industrial automation. Its ability to provide location-based services makes it a valuable tool for many applications, and its low power consumption makes it an ideal solution for IoT devices and other battery-powered applications[8].

5. BLE advantages and disadvantages

Every technology has its strong and weak sides. For the BLE technology it is possible to specify the next advantages:

Low Power Consumption: One of the biggest advantages of BLE is its low power consumption. BLE devices can operate for years on a single coin cell battery, making them ideal for small, low-power devices such as wearables, sensors, and other IoT devices.

Low Cost: BLE technology is low-cost, making it an affordable solution for manufacturers and consumers. It has lower hardware costs compared to traditional Bluetooth technology and other wireless technologies.

Easy Integration: BLE technology can easily integrate with other wireless technologies, such as Wi-Fi and NFC. This makes it easier for devices to communicate with each other and with the internet[9].

Robustness: BLE technology is designed to operate in noisy radio frequency environments, making it robust and reliable even in the presence of interference from other wireless devices.

Security: BLE technology has several layers of security built into it, including data encryption, user authentication, and secure pairing. This makes it a safe and secure technology for transmitting sensitive data.

However the Bluetooth Low Energy technology has some disadvantages that needs to be solved or taken to the account while using it:

Limited Range: The range of BLE devices is limited to around 10-30 meters, depending on the environment. This is a disadvantage when compared to other wireless technologies such as Wi-Fi, which can cover larger distances.

Limited Bandwidth: BLE technology has a limited bandwidth, which means it can only transmit small amounts of data at a time. This makes it unsuitable for applications that require high-speed data transfer[10].

Compatibility Issues: BLE technology is not compatible with older devices that use traditional Bluetooth technology. This can be a disadvantage if older devices need to communicate with newer BLE devices.

Complexity: BLE technology is more complex to set up and configure compared to traditional Bluetooth technology. This can be a disadvantage for non-technical users who may find it difficult to use.

6. Ultra-Wide Band technology overview

Ultra-wideband (UWB) technology is a wireless communication protocol that utilizes very short-duration pulses of electromagnetic waves to transmit data over a wide frequency spectrum. UWB signals occupy a frequency bandwidth that is typically several GHz wide, compared to the narrowband signals used in traditional wireless communication systems, such as Wi-Fi and Bluetooth[11].

UWB technology is designed to enable high-speed wireless data transfer, accurate location tracking, and other applications that require high bandwidth and low power consumption. UWB technology was originally developed for military and scientific applications in the 1960s, and it was later used in commercial applications such as radar systems and ground-penetrating radars. In the early 2000s, UWB technology was standardized by the Federal Communications Commission (FCC) in the United States, which opened the door for its use in commercial applications.

One of the key advantages of UWB technology is its ability to accurately measure the distance between two UWB-enabled devices. This is made possible by the short duration of the UWB pulses, which allows for precise time-of-flight measurements of the signal. UWB distance measurement is highly accurate, with a typical error of only a few centimeters, making it ideal for applications such as indoor location tracking and asset tracking[12].

Another advantage of UWB technology is its high data transfer rates. UWB can achieve data transfer rates of up to 480 Mbps, which is significantly higher than Wi-Fi and other wireless communication protocols. This makes UWB ideal for high-bandwidth applications such as streaming video and audio. UWB technology also offers low power consumption, making it suitable for battery-powered devices such as smartphones, wearables, and Internet of Things (IoT) devices. UWB devices can operate in a low-power mode when not transmitting data, which helps to conserve battery life.

One of the main challenges of UWB technology is its limited range. UWB signals have a shorter range than Wi-Fi and other wireless communication protocols, typically only a few meters. This makes UWB ideal for short-range applications such as indoor location tracking and asset tracking, but less suitable for long-range applications such as outdoor communication[13].

Another challenge of UWB technology is its susceptibility to interference from other wireless devices. UWB signals can be affected by other wireless signals operating in the same frequency band, which can lead to reduced signal quality and data transfer rates.

Despite these challenges, UWB technology has gained traction in recent years, particularly in the areas of indoor location tracking and asset tracking. UWB-enabled devices are becoming increasingly common in smartphones,

wearables, and other consumer electronics, and the technology is being adopted in various industrial and commercial applications, such as automotive and logistics.

7. UWB Advantages and disadvantages

Ultra-Wide Band is the good positioning technology that has the next strong sides:

High Precision: Ultra-Wide Band (UWB) technology offers high precision location accuracy, which makes it an ideal choice for applications such as indoor navigation, asset tracking, and industrial automation. UWB can achieve sub-centimeter accuracy, which is much better than other wireless technologies such as Wi-Fi or Bluetooth.

Immunity to Interference: UWB operates on a wide frequency band, which allows it to avoid interference from other wireless signals such as Wi-Fi or Bluetooth. This makes UWB a reliable and robust technology for use in harsh and noisy environments[14].

High Data Rates: UWB technology can achieve very high data rates of up to 10 Gbps, which is much faster than other wireless technologies such as Wi-Fi or Bluetooth. This makes UWB an ideal choice for high-bandwidth applications such as streaming video or audio.

Low Power Consumption: UWB devices require minimal power, which makes them ideal for use in battery-operated devices such as smartphones, smartwatches, and IoT sensors. This enables UWB devices to operate for longer periods of time without needing a battery recharge.

Security: UWB technology provides a high level of security through its use of short duration pulses, which makes it difficult for unauthorized users to intercept or decode the transmitted data.

In order to these strong sides UWB has some challenges and weak sides that need to be mentioned:

Cost: UWB technology is still relatively expensive compared to other wireless technologies such as Wi-Fi or Bluetooth. This can make it challenging for businesses to justify the cost of implementing UWB technology in their operations.

Limited Range: UWB technology has a limited range of typically around 10-20 meters, which makes it unsuitable for applications that require longer range communication such as outdoor navigation[15].

Line-of-Sight Requirement: UWB signals are highly directional and require a clear line-of-sight between the transmitter and receiver for optimal performance. This can limit the practical applications of UWB technology, especially in environments where obstacles such as walls or furniture may interfere with the signal.

Limited Availability: UWB technology is not yet widely adopted, which can limit its availability and interoperability with other devices and systems.

Regulatory Restrictions: UWB technology is subject to regulatory restrictions in some countries, which can limit its adoption and deployment in certain regions.

Conclusion

In conclusion, each wireless technology - Wi-Fi, BLE, and UWB - has its own strengths and weaknesses, making it suitable for specific applications. Wi-Fi offers a wide range and high data rates, making it ideal for applications such as internet connectivity and large-scale asset tracking. BLE is a low-power technology with good range and is well-suited for applications such as indoor navigation and proximity sensing. UWB technology offers high precision location accuracy, low power consumption, and immunity to interference, making it an ideal choice for applications such as indoor asset tracking, industrial automation, and secure communication.

The continued evolution and development of wireless technologies hold tremendous potential for transforming industries and enabling new applications. As the demand for wireless connectivity and location services continues to grow, it is likely that these technologies will continue to evolve and improve, offering new opportunities and solutions for a wide range of industries and applications.

References:

- [1] *Youssef M., Agrawala A.*, The Horus WLAN Location Determination System. // 2007 - pp. 205-218.
- [2] *Liu Y., Chen Y., Wang H., Lu, J.*, Wi-Fi Indoor Positioning: Recent Advances and Comparisons. // 2017 - pp. 801-824.
- [3] *Li Y., Li X.*, A comprehensive Survey of Wi-Fi Positioning Technology. // 2014 - pp. 1-19.
- [4] *Brena R. F., Calderon J.A., Villasenor L.*, A Survey of Wi-Fi-Based Indoor Positioning Systems. // 2019 - pp. 1-27
- [5] *Wang C., Chen Y., Zhang H.*, Wi-Fi Fingerprint Indoor Positioning Based on Convolutional Neural Network. // 2019 - pp. 87304-87314
- [6] *Kumar P., Kaur S.*, Bluetooth Low Energy: A review. // 2019 - pp. 678-683
- [7] *Adhikari B., Lai C.F.*, Bluetooth Low Energy: A Technology Overview. // 2016 - pp. 654-682
- [8] *Lui H., Chen Y., Chen W.*, A survey on Bluetooth Low Energy: Architecture, Applications and Performance Issues.// 2019 - pp. 85368 - 85388
- [9] *Perez A.A., Ortiz J.*, Bluetooth Low Energy for IoT: Applications, Trends and Future Directions // 2019 - pp.695-707
- [10] *Hekmat R., Abid M.*, A comprehensive Study of Bluetooth Low Energy (BLE) in IoT environment. //2018 - pp. 44-51
- [11] *Zafari F., Fort A.*, Ultra-Wideband Indoor Positioning Systems: A review. // 2019 - pp. 8593-8606
- [12] *Anwar S., Kim D.*, A Comprehensive Review on Ultra-Wideband Positioning Techniques for Wireless Sensor Networks. // 2019 - p. 2334
- [13] *Nikaein N., Lunden P.*, Ultra-Wideband for 5G and IoT: A Tutorial. // 2018 - pp. 162-168
- [14] *Zhang X., Yan Y., Wang Y.*, A Review of Ultra-Wideband Positioning Systems. // 2018 - pp 1-16
- [15] *Guo H., Wang W.*, Performance Evaluation of Ultra-Wideband Indoor Positioning Systems // 2019 - pp 509-513/

The article was delivered to editorial staff on the 18.10.2022



С.Ф. Чалий¹, В.О. Лещинський²

¹професор кафедри інформаційних управляючих систем,
Харківський національний університет радіоелектроніки, Україна,
serhii.chalyi@nure.ua

²доцент кафедри програмної інженерії,
Харківський національний університет радіоелектроніки, Україна,
volodymyr.leshchynskyi@nure.ua

МОЖЛИВІСНО-КАУЗАЛЬНЕ ПРЕДСТАВЛЕННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНІЙ ІНФОРМАЦІЙНІЙ СИСТЕМІ

Предметом вивчення в статті є процеси побудови пояснень щодо отриманих результатів та послідовності дій з прийняття рішення в інтелектуальній інформаційній системі. Метою є розробка підходу до побудови пояснень на основі можливісного опису причинно-наслідкових зв'язків між вхідними даними та результатом роботи інтелектуальної системи, що створює умови для формування пояснення при представленні такої системи як у вигляді «чорного», так і у вигляді «сірого» ящика. Завдання: розробка узагальненої можливісно-каузальної моделі пояснення; розробка методу побудови можливісно-каузального представлення пояснення в інтелектуальній інформаційній системі. Використовуваними підходами є: методи побудови пояснень, теорія можливостей, підходи до побудови темпоральних знань. Висновки. Наукова новизна отриманих результатів полягає в наступному. Запропоновано комплекс можливісно-каузальних моделей пояснення в інтелектуальній системі, що задає причинно-наслідковий зв'язок між класом вхідних даних та класом рішення, а також між проміжними діями з процесу отримання результату та класом рішення. Можливісний аспект моделі розраховується для підмножини, що містить представників одного класу даних, або ж для окремих дій спрощеного процесу прийняття рішення. У практичному плані розроблена модель дає можливість сформулювати опис процесу прийняття рішення на основі можливісних каузальних залежностей та побудувати пояснення на основі обмежених даних про процес функціонування інтелектуальної інформаційної системи. Запропоновано метод побудови пояснення в інтелектуальній інформаційній системі на основі можливісних каузальних залежностей. Метод містить етапи визначення класів вхідних даних та результату для пояснення, формування переліку можливих причинно-наслідкових залежностей, що пов'язують вхідні дані або дії процесу із рішенням інтелектуальної системи, розрахунку можливостей використання отриманих залежностей для побудови пояснення, розрахунку необхідності для отриманих залежностей та упорядкування отриманих пояснень за значенням необхідності. Метод дає можливість побудувати пояснення при представленні інтелектуальної інформаційної системи як у вигляді «чорного», так і у вигляді «сірого» ящика, відобразивши відповідно вплив вхідних даних та вплив спрощеного процесу прийняття рішення на результат інтелектуальної системи.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА, СИСТЕМА ШТУЧНОГО ІНТЕЛЕКТУ, ПОЯСНЕННЯ, ПРОЦЕС ПРИЙНЯТТЯ РІШЕНЬ, ПРИЧИННО-НАСЛІДКОВИЙ ЗВ'ЯЗОК, МОЖЛИВІСТЬ, КАУЗАЛЬНІСТЬ

S. Chalyi, V. Leshchynskyi Possibility-causal representation of explanations in an intelligent information system.

The article's subject matter is the process of constructing explanations for the obtained results and the sequence of decision-making actions in the intelligent information system. The goal is to develop an approach to the construction of explanations based on a possible description of cause-and-effect relationships between input data and the result of the work of an intelligent system, which creates conditions for the formation of an explanation when presenting such a system both in the form of "black" and in the form of "gray" box. Tasks: development of a generalized possible-causal model of explanation; development of a method of possible-causal representation of an explanation in an intellectual information system. The used approaches are: methods of constructing explanations, the theory of possibilities, approaches to the construction of temporal knowledge. Conclusions. The scientific novelty of the obtained results is as follows. A possible-causal model of explanation in an intelligent system is proposed, which specifies a cause-and-effect relationship between the class of input data and the class of decision, as well as between intermediate actions from the process of obtaining a result and the class of decision. The probabilistic aspect of the model is calculated for a subset containing representatives of the same class of data, or for individual actions of a simplified decision-making process. In practical terms, the developed model makes it possible to form a description of the decision-making process based on possible causal dependencies and to build an explanation based on limited data about the process of functioning of the intelligent information system. A method of constructing an explanation in an intelligent information system based on possible causal dependencies is proposed. The method includes the stages of determining the classes of input data and the result for explanation, forming a list of possible cause-and-effect dependencies that connect input data or process actions with the decision of an intelligent system, calculating the possibilities of using the obtained dependencies to build an explanation, calculating the need for the obtained dependencies and ordering received explanations according to necessity. The method makes it possible to build an explanation when presenting an intelligent information system both in the form of a "black" and in the form of a "gray" box, reflecting, respectively, the influence of input data and the influence of a simplified decision-making process on the result of an intelligent system.

INTELLIGENT SYSTEM, ARTIFICIAL INTELLIGENCE SYSTEM, EXPLANATION, DECISION-MAKING PROCESS, CAUSALITY, POSSIBILITY, CAUSALITY

Вступ

Сучасні дослідження в галузі психології пізнання свідчать про те, що людина, як правило, потребує обґрунтування нових знань з використанням відповідних пояснень [1-3].

Пояснення щодо процесу прийняття рішення, які надаються в інтелектуальних системах, формують причинно-наслідкові зв'язки між вхідними даними та результатом [2] і тому забезпечують умови для того, щоб користувачі були впевнені у правильності отриманих результатів [4, 5].

Актуальність використання пояснень в інтелектуальних системах є наслідком протиріччя, що виникає при використанні алгоритмів машинного навчання у процесі формування рішень. Такі алгоритми орієнтовані на представлення таких систем у вигляді «чорного ящика», що робить їх незрозумілими для користувача. До того ж результати, отримані при використанні алгоритмів машинного навчання, можуть бути спотворені внаслідок упередженості або викидів у вхідних даних. Відповідно, отримані в інтелектуальній інформаційній системі рішення можуть не в повній мірі задовільнити користувачів таких систем. На практиці така невідповідність приводить до неефективного використання інтелектуальної інформаційної системи, зокрема до відмови від застосування запропонованих рішень.

При використанні «прозорих» алгоритмів формування рішення виникає інше обмеження, пов'язане із юридичним захистом інтелектуальної власності. В даному випадку має місце заборона щодо розкриття деталей та умов процесу прийняття рішення для користувачів. Юридичні обмеження також призводять до зниження довіри користувачів до рішень системи штучного інтелекту та відповідного неефективного використання цих рішень на практиці [6].

Тому наукові дослідження в області формування пояснень для інтелектуальних систем інтенсивно розвиваються в останні роки, особливо в рамках програми XAI, що була започаткована керівництвом агенції DARPA у 2017 році [7].

Ключові напрямки досліджень у сфері пояснюваних інтелектуальних систем базуються як на визначенні відповідності пояснень потребами користувачів, наприклад користувачів рекомендаційних систем [8], так і на визначенні явних або неявних причинно-наслідкових залежностей між вхідними факторами та рішенням інтелектуальної системи [9-13].

Однак існуючі підходи є спеціалізованими, значною мірою залежать від типу алгоритму, що використовується в інтелектуальній системі, і не приділяють достатньо уваги побудові пояснень на основі комбінованого опису процесу прийняття рішення, що враховує як зв'язок між вхідними даними та результатом інтелектуальної системи, так і залежності між

ключовими діями та результатом вказаного процесу. Такий зв'язок носить можливісний характер, оскільки він враховує не лише можливість впливу вхідних даних на рішення системи, а й необхідність вибору підмножини цих вхідних даних, які є найбільш суттєвими для пояснення щодо результату роботи інтелектуальної інформаційної системи. Можливісний підхід [14] дає можливість узагальнено описати каузальні залежності для побудови пояснень, абстрагуючись від особливостей конкретного механізму прийняття рішень в інтелектуальній системі.

Зазначене свідчить про актуальність задачі побудови каузального представлення пояснень з урахуванням можливісного опису такого причинно-наслідкового зв'язку.

1. Постановка задачі

Метою статті є розробка підходу до побудови пояснень на основі можливісного опису причинно-наслідкових зв'язків між вхідними даними та результатом роботи інтелектуальної системи, що створює умови для формування пояснення при представленні такої системи як у вигляді «чорного», так і у вигляді «сірого» ящика.

Для досягнення поставленої мети вирішуються такі задачі:

- розробка узагальненої можливісно-каузальної моделі пояснення;
- розробка методу побудови можливісно-каузального представлення пояснення в інтелектуальній інформаційній системі.

2. Схема побудови пояснення в системі штучного інтелекту

Загальна схема побудови пояснення у існуючій системі штучного інтелекту, що представлена у вигляді «чорного» або «сірого» ящика, полягає у формуванні зовнішньої пояснювальної підсистеми. Дана підсистема використовує вхідні дані, вихідний результат, а також доступні проміжні дані про процес прийняття рішення у системі штучного інтелекту. Для пояснення отриманого результату використовується спрощена модель прийняття рішення.

При представленні системи штучного інтелекту у вигляді «чорного ящика» доступними для побудови пояснення є лише вхідні дані та рішення системи. Відповідну схему побудови пояснення представлено на рис. 1.

В даному випадку формуються причинно-наслідкові залежності між значеннями окремих елементів вхідних даних та рішенням. Ключова ідея пояснення полягає в тому, щоб показати залежності, що відображають вплив ключових значень вхідних даних на отримане рішення.

З метою спрощення моделі процесу прийняття рішення при побудові пояснення доцільно

визначити залежності не між окремими вхідними даними та отриманим результатом, а між класами елементів вхідного набору даних та класом рішення.

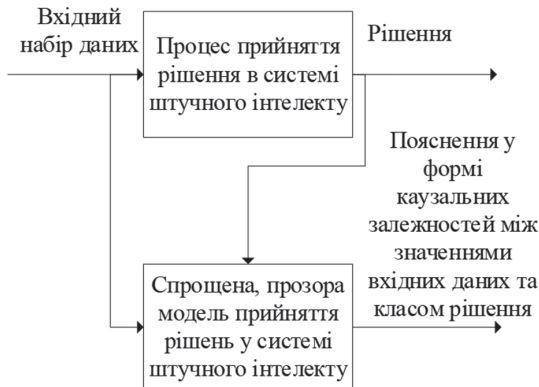


Рис. 1. Побудова пояснення на основі вхідних даних при представленні системи штучного інтелекту у вигляді «чорного ящика»

Наприклад, пояснення щодо результату класифікації зображення людини, що біжить, може відображати залежності між фрагментами зображення, які містять елементи тіла в русі, та результуючим рішенням. Або пояснення щодо пропозиції комп'ютера в рекомендаційній системі може містити зв'язки між значенням комплектуючих (модель процесора, об'єм пам'яті, тощо) та запропонованою моделлю комп'ютера.

Слід зауважити, що вказані залежності формуються для груп комп'ютерів. Наприклад, процесор типу i7 може виступати в якості причини вибору ноутбука певної фірми внаслідок того, що забезпечується найкраще співвідношення ціни та потужності.

При представленні системи штучного інтелекту у вигляді «сірого ящика» доступною є також частина інформація про процес прийняття рішення у такій системі. Дана інформація зазвичай представлена у формі логу (журналу подій). Останній формується підсистемою моніторингу.

Відповідна схема побудови пояснення представлена на рис. 2.

Журнал подій S містить у собі набір трас $\{S_m\}$.

Кожна з трас S_m описує доступну для зовнішнього спостерігача послідовність станів процесу прийняття рішення:

$$S_m = \langle s_{m,1}, s_{m,2}, \dots, s_{m,z}, \dots \rangle \quad (1)$$

Стани задаються через множину значень змінних, що характеризують властивості системи. Стани є упорядкованими у часі, оскільки містять темпоральні мітки.

Тобто кожний стан описується множиною значень змінних $s_{m,z} = \{x_{m,z}^i, t_{m,z}^i\}$, серед яких $x_{m,z}^i$ задає значення таких, наприклад, властивостей, як:

– назва дії із процесу прийняття рішення, що привела до поточного стану;

– об'єкт або змінна, з якою виконувалась дія, тощо.

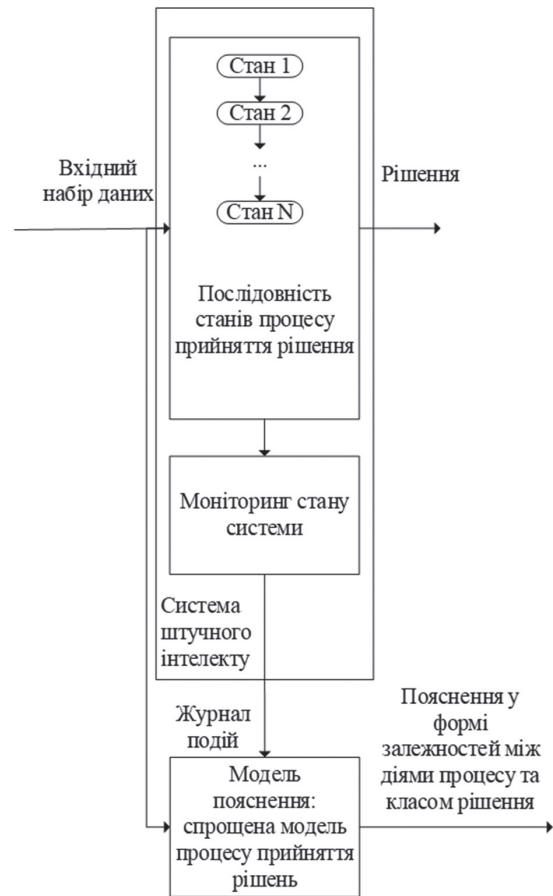


Рис. 2. Побудова пояснення на основі журналу подій при представленні системи штучного інтелекту у вигляді «сірого ящика»

Змінна $t_{m,z}^i$ є позначкою часу, яка дає можливість задати темпоральні правила.

Характерна особливість логу полягає в тому, що кожна з трас є записом про виконання одного екземпляру процесу прийняття рішення, а сукупність трас задає всі реалізовані на практиці альтернативи такого процесу.

Відповідно, зворотний інжиніринг логу дає можливість побудувати модель процесу і, на основі аналізу такої моделі, знайти причини отриманого в системі штучного інтелекту рішення. Такий зворотний інжиніринг зазвичай виконується методами інтелектуального аналізу процесів.

Однак дослідження отриманої в результаті інтелектуального аналізу процесів моделі потребує додаткової підготовки, що утруднює сприйняття такого пояснення користувачем.

Альтернативний підхід до побудови пояснення базується на виділенні темпоральних залежностей між станами системи. Такі залежності представляються у формі темпоральних правил. Темпоральні задають порядок пар станів $s_{m,z}$ у часі:

Темпоральні правила мають вигляд:

$$fn : s_{m,z} \rightarrow s_{m,z+1}, \quad (2)$$

$$ff : s_{m,z} \rightarrow s_{m,z+q} | q > 1. \quad (3)$$

Правило (1) визначає зв'язок для двох послідовних у часі станів процесу прийняття рішення.

Правило (2) задає зв'язок для двох станів $s_{m,z}$ та $s_{m,z+q}$, між якими є проміжні у часі стани, наприклад $s_{m,q-1}$.

Наведені правила дають можливість відобразити як поточні (1), так і довгострокові (2) наслідки дій процесу прийняття рішення в інтелектуальній інформаційній системі.

Узагальнене правило f_q^m об'єднує правила (2) та (3) та упорядковує у часі довільну пару станів $s_{m,z}$ та $s_{m,z+q}$:

$$f_q^m : s_{m,z} \rightarrow s_{m,z+q} | q \geq 1. \quad (4)$$

Досягнення результату роботи системи типу R_l можна розглядати як кінцевий стан $s_{m,Q}$ процесу прийняття рішення. Відповідно, опис процесу у вигляді темпоральних правил дає можливість використати для побудови пояснення темпоральні залежності між ключовими проміжними станами системи штучного інтелекту та отриманим рішенням (5), або ж між патерном послідовності станів та відповідним рішенням (6).

$$s_{m,z} \rightarrow R_l, \quad (5)$$

$$f_q^z \rightarrow R_l. \quad (6)$$

При необхідності деталізувати стани дані залежності можуть бути зведені до зв'язку між доступними значеннями проміжних змінних системи $x_{m,z}^i$ та отриманим рішенням, або ж між послідовністю зміни значень таких змінних та результатом.

3. Можливісно-каузальна модель пояснення в інтелектуальній інформаційній системі

Задача представлення каузальних залежностей для пояснень у випадку представлення інтелектуальної системи як «чорного ящика» полягає у встановленні зв'язку між класами вхідних даних та класами рішень.

Позначимо множину вхідних даних як $X = \{x_i\}$.

Вхідні дані підрозділяються на класи, які описуються підмножинами X_j :

$$X_j \subset X, (\cap X_j) = \emptyset. \quad (7)$$

Аналогічно, позначимо множину рішень інтелектуальної інформаційної системи $R = \{r_i\}$. Класи рішень визначаються підмножинами R_l , що не перетинаються:

$$R_l \subset R, (\cap R_l) = \emptyset \quad (8)$$

Для кожного з рішень зазвичай є відомою ймовірність використання вхідних даних. Наприклад, якщо було продано ноутбук в системі електронної

комерції, то відомими є комплектуючі цього ноутбука. Відповідно, можна підрахувати ймовірність вибору ноутбуку з певним процесором, жорстким диском або пам'яттю. Таку множину ймовірностей для елементів X_j позначимо P_j .

Тобто $P_j = \{p_{k,j}\}$, причому $p_{k,j}$ – це ймовірність вибору рішення R_l із вхідними даними x_i .

Тоді можливісно-каузальна модель пояснення при представленні інтелектуальної системи у вигляді «чорного ящика» для однієї вхідної змінної має вигляд:

$$\text{Вхідні дані} : \{P_j\}, \{R_l\},$$

$$\text{Можливісно – каузальна залежність} : \quad (9)$$

$$x_{k,j} \Rightarrow R_l | p_{k,j} = \Pi(X_j),$$

де $x_{k,j}$ – значення k – вхідної змінної із підмножини X_j , яка є можливою причиною рішення класу R_l ; $\Pi(X_j)$ – можливість використання одного із елементів підмножини X_j в якості причини для рішення класу R_l .

Сенс виразу (2) полягає в наступному: значення $x_{k,j}$ X_j змінної із підмножини можливих значень X_j є найбільш можливою причиною отриманого рішення.

Модель пояснення для декількох вхідних змінних має такий вигляд:

$$\text{Вхідні дані} : \{P_j\}, \{R_l\},$$

$$\text{Можливісно – каузальна залежність} :$$

$$\langle x_{k,j}, x_{k,j+1}, \dots \rangle \Rightarrow R_l \quad (10)$$

$$| p_{k,j} = \Pi(X_j), p_{k,j+1} = \Pi(X_{j+1}),$$

$$N(X_j) \geq N(X_{j+1})$$

В даному випадку вхідні змінні упорядковуються за необхідністю, оскільки необхідність фактично визначає можливість використання альтернативних до $x_{k,j}$ значень вхідних даних для пояснення.

У випадку представлення інтелектуальної системи як «сірий ящик» доступною є додаткова інформація у вигляді логів або журналів подій, що дає можливість побудувати опис відомої частини процесу прийняття рішення у вигляді темпоральних правил. Тоді модель пояснення може базуватись на трьох підходах:

– на основі обмежень на виконання дій процесу прийняття рішення;

– на основі вибору можливої дії/стану процесу прийняття рішення серед підмножини альтернатив;

– на основі вибору можливого темпорального правила.

Згідно першого підходу, в якості можливої причини рішення розглядаються обмеження на виконання процесу прийняття рішення в інтелектуальній системі. В якості обмежень розглядаються дії, які були виконані на всіх трасах процесу. Тобто для обмежень

виконується умова $p_{m,q} = 1$. Відповідна можливісно-каузальна залежність задає зв'язок між станами $s_{m,q}$ та класом рішення R_i .

Згідно другого підходу в якості можливої причини рішення інтелектуальної системи розглядається один із альтернативних станів процесу прийняття рішення. Тобто із станів на різних трасах формується множина альтернатив і як причина рішення вибирається стан із найбільшою ймовірністю. Представлення такої залежності має вигляд:

$$\begin{aligned} & \text{Вхідні дані: } \{P_j\}, \{R_i\}, \\ & \text{Можливісно-каузальна залежність:} \\ & \langle s_{m,z}, s_{m,q}, \dots \rangle \Rightarrow R_i \quad (11) \\ & | p_{m,z} = \Pi(S_z), p_{m,q} = \Pi(S_q), \\ & N(S_z) \geq N(S_q), \end{aligned}$$

де S_z, S_q – підмножини альтернативних станів процесу прийняття рішення.

В даному випадку фактично конструюються темпоральні правила $s_{m,z} \rightarrow R_i, s_{m,q} \rightarrow R_i$, на базі яких визначаються можливісні каузальні залежності результату від відповідних дій процесу прийняття рішення. Зв'язок між станами та результатом у наведених правилах виділяється тому, що проміжні стани відображають результати окремих дій у процесі формування рішення.

Згідно третього підходу, формується зв'язок між темпоральними правилами та результатом. Тобто задана правилами послідовність дій розглядається як можлива причина отриманого в інтелектуальній системі результату.

Розглянемо приклад залежності між проміжними станами процесу прийняття рішення та результатом для рекомендаційної системи. В даному випадку в онлайн-підсистемі побудови рекомендацій аналізується послідовність дій користувача системи (тобто послідовність кліків по екранній формі). Користувачі по різному взаємодіють з однією й тією ж екранною формою, вибираючи або пропускаючи запропоновані на екрані товари. Тому аналіз трас логу для різних користувачів дає можливість порівняти траєкторії їх руху та надати один із варіантів переміщення по екрану як пояснення щодо запропонованої рекомендації товарів або послуг. Зокрема, якщо користувач розглядає (вибирає мишою) властивості найбільш популярних товарів, то відповідна залежність буде використана у якості пояснення.

4. Метод побудови можливісно-каузального представлення пояснення в інтелектуальній інформаційній системі

Побудова неведеного можливісно-каузального представлення передбачає побудову опису вхідних даних, формування можливих залежностей та

подальше їх упорядкування з урахуванням розрахованої необхідності впливу конкретних значень вхідних змінних на отримане в системі штучного інтелекту рішення.

Метод містить таку узагальнену послідовність етапів.

Етап 1. Визначення класів вхідних та вихідних даних для пояснення.

Крок 1.1 Визначення класів вхідних даних.

Реалізація даного етапу залежить від особливостей та структури вхідних даних. Однак в цілому має бути отримана однорівнева (по групам) або багаторівнева (ієрархічна) класифікація вхідних даних.

Результатом даного етапу є множина класів X_j , що описується підмножинами X_j . Для однорівневого розбиття на класи виконується умова (7). При побудові ієрархії класів вхідних даних вираз (7) доповнюється рекурсивно, тобто виділяються підмножини $X_{j,g}$, для яких виконується умова:

$$\bigcup_g X_{j,g} = X_j, \left(\bigcap_g X_{j,g} \right) = \emptyset. \quad (12)$$

Виділення підмножин інших рівнів ієрархії виконується аналогічно.

Крок 1.2 Формування класів рішень.

На даному етапі формуються класи рішень таким чином, щоб для кожного рішення з класу можна було сформулювати ідентичне або схоже пояснення.

Класи рішень R_i визначаються так, щоб любе рішення даного класу мало множину значень вхідних даних із відповідних класів.

Наприклад, якщо рішенням є пропозиція ноутбука в рекомендаційній системі, то клас рішення містить всі ноутбуки з однаковим типом процесору, об'ємом пам'яті та жорсткого диску, тощо. Тобто клас рішення R_i визначається одними й тими ж підмножинами X_j , для яких ми можемо розрахувати можливісно-каузальні залежності:

$$X^l = \{ X_j : \forall x_i \in X_j \exists r_n \in R_i \} \quad (13)$$

Крок 1.3 Визначення класів темпоральних правил. Даний крок виконується за умови, що інтелектуальна система представлена як «сірий ящик».

На даному кроці темпоральні правила формуються та розподіляються на класи в залежності від значень атрибутів $x_{m,z}^i$ станів $s_{m,z}$.

Тоді клас темпорального правила визначається, наприклад, класом $x_{m,z}^i$. Зокрема, ми можемо виділити дії за їх функціональною ознакою: фільтрація, перевірка умов, виявлення латентних факторів, тощо.

Етап 2. Формування можливих причинно-наслідкових залежностей.

На даному етапі виділяються залежності виду $x_{k,j} \Rightarrow R_i$ для вхідних змінних та $s_{m,z} \Rightarrow R_i$ й $f_q^z \Rightarrow R_i$ для темпоральних правил.

Етап 3. Розрахунок можливостей використання отриманих залежностей для побудови пояснення.

Можливість розраховується як найбільша ймовірність $p_{k,j}$ для відповідної підмножини вхідних даних, станів або правил.

Етап 4. Розрахунок необхідності для отриманих залежностей.

Необхідність розраховується через можливість всіх інших значень змінних, крім змінних із підмножини X_j .

Етап 5. Упорядкування отриманих пояснень за значенням необхідності.

Результатом даного етапу є набір змінних, які є можливими причинами отриманого рішення. Ці змінні упорядковуються за ступенем довіри до них. Остання визначається згідно необхідності використання цих змінних.

В тому випадку, якщо інтелектуальна система представлена у вигляді «сірого ящика», тоді результатом є набір темпоральних правил, які є можливими причинами рішення із класу R_l .

Особливість використання темпоральних правил полягає в тому, що вони визначають не лише проміжні стани, які є можливими причинами для отриманого результату, але й дії, які привели до цих станів, а також зв'язок між діями, тобто упорядкованість дій-причини та попередньої до неї дії в часі.

В результаті можна отримати комплексну причину результату: ключова дія (або дії), які привели до даного рішення, а також передумови для виконання цих ключових дій.

Так, якщо ми розглядаємо людино-машинний процес прийняття рішення, при якому людина використовує проміжне рішення інтелектуальної системи і відповідно змінює алгоритм роботи, то в якості причини ми можемо з'ясувати, чи використала ця людина-оператор проміжні результати роботи системи на подальших етапах процесу прийняття рішення.

Наприклад, чи були використані результати автоматизованої діагностики у процесі сервісного обслуговування клієнтів. Або ж навпаки, з метою спрощення роботи виконавець проігнорував отриману інформацію.

Розглянемо приклад формування можливісно-каузальної залежності для пояснення щодо пропозиції комп'ютера в рекомендаційній системі.

Використовуються вхідні дані про продажі ноутбуків у системі електронної комерції. Ці містять інформацію про модель комп'ютера.

На базі інформації про моделі ноутбуків формується множина класів рішень $R = \{R_l\}$, що містить моделі ноутбуків на базі процесорів $i7, i9$, тощо.

Множина класів вхідних даних, для, наприклад, процесора, формується у вигляді набору $X = \{i3, i5, i7, i9\}$. Кожна із підмножин $i3, i5, i7, i9$

містить назви моделей відповідного типу, наприклад $i7 = \{i7-1185, i7-1165, \dots\}$.

На базі інформації про продажі формується множина ймовірностей продажу ноутбуків із відповідними процесорами: $p(i7) = \{p(i7-1185), p(i7-1165), \dots\}$.

На наступних етапах розраховується можливість $\Pi(i7)$:

$$\Pi(i7) = \max(p(i7-1185), p(i7-1165), \dots), \quad (14)$$

та формується каузальна залежність виду

$$i7 - 1185 \Rightarrow \text{Модель ноутбука} \quad (15)$$

як основа пояснення щодо запропонованої моделі.

Аналогічні розрахунки виконуються для інших вхідних змінних, пов'язаних із пам'яттю, жорстким диском, параметрами екрану, зовнішніх портів, тощо.

Залежності виду (15) в подальшому упорядковуються за значенням необхідності.

Висновки

Розроблено комплекс можливісно-каузальних моделей пояснення в інтелектуальній системі, що задають причинно-наслідковий зв'язок між класами вхідних даних, станів, темпоральних правил та класом рішення.

При представленні інтелектуальної системи у вигляді «чорного ящика» пояснення визначається через можливість впливу вхідних даних на рішення інтелектуальної системи. Можливість розраховується для підмножини, що містить представників одного класу даних.

Модель в даному випадку забезпечує пояснення щодо рішення інтелектуальної системи через вплив вхідних даних на результат, представляючи такий вплив як можливісний каузальний зв'язок між входом та виходом інтелектуальної системи.

При представленні інтелектуальної системи у вигляді «сірого ящика» пояснення визначається через причинно-наслідкові зв'язки між діями спрощеного процесу прийняття рішення.

Безпосередньо спрощений процес відображає послідовність отримання результату, сформовану на основі доступних даних про функціонування інтелектуальної системи, наприклад, на основі журналів подій. Така послідовність представляється темпоральними залежностями.

Можливісний аспект моделі у даному випадку визначається для окремих дій або послідовностей дій спрощеного процесу прийняття рішення.

У практичному плані розроблена модель орієнтована на формування опису процесу прийняття рішення з використанням можливісного підходу, що створює умови для побудови зрозумілого опису процесу прийняття рішення у вигляді множини взаємопов'язаних можливісних каузальних залежностей на основі обмежених даних про процес

функціонування інтелектуальної інформаційної системи.

Розроблено метод побудови пояснення в інтелектуальній інформаційній системі на основі можливих каузальних залежностей.

Метод містить етапи визначення класів вхідних даних для пояснення, формування можливих причинно-наслідкових залежностей, розрахунку можливостей використання отриманих залежностей для побудови пояснення, розрахунку необхідності для отриманих залежностей та упорядкування отриманих пояснень за значенням необхідності.

Метод дає можливість побудувати пояснення при представленні інтелектуальної інформаційної системи як у вигляді «чорного», так і у вигляді «сірого» ящика, відобразивши відповідно вплив вхідних даних та вплив спрощеного процесу прийняття рішення на результат інтелектуальної системи.

Список літератури:

- [1] *Chi, M., de Leeuw, N., Chiu, M., & LaVancher, C.* Eliciting self-explanations improves understanding. *Cognitive Science*. – 1994. – Vol.18. P. 439–477.
- [2] *Чалий, С., & Лещинська, І.* Концептуальна ментальна модель пояснення в системі штучного інтелекту // Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології – 2023. – Vol. 1(9). – P. 70–75 <https://doi.org/10.20998/2079-0023.2023.01.11>
- [3] *Carey, S.* The origin of concepts. New York, NY: Oxford University Press. 2009. 608 p.
- [4] *Adadi, A., Berrada, M.* (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access*. – 2018. – Vol.6. P. 52138– 52160.
- [5] *Castelvecchi D.* Can we open the black box of AI? *Nature*. – 2016. – Vol. 538 (7623), pp. 20-23.
- [6] *Tintarev N., Masthoff J.* Evaluating the effectiveness of explanations for recommender systems, *User Model User-Adap Inter*. – 2012. – Vol. 22, pp. 399– 439, <https://doi.org/10.1007/s11257-011-9117-5>
- [7] *Gunning I. D. Aha* DARPA’s Explainable Artificial Intelligence (XAI) Program // *AI Magazine*. – 2019. – Vol. 40(2), pp.44-58, doi: 10.1609/aimag.v40i2.2850.
- [8] *Adomavicius G. et al.* Incorporating contextual information in recommender systems using a multidimensional approach // *ACM Transactions on Information Systems*. – 2005. – Vol. 23(1). – P. 103–145.
- [9] *Chalyi, S., Leshchynskiy, V.* Method of constructing explanations for recommender systems based on the temporal dynamics of user preferences. *EUREKA: Physics and Engineering*. – 2020. – Vol. 3, pp. 43-50. doi: 10.21303/2461-4262.2020.001228. Available at: <http://journal.eu-jr.eu/engineering/article/view/14>.
- [10] *Chalyi S.* Probabilistic counterfactual causal model for a single input variable in explainability task / S. Chalyi, V. Leshchynskiy // *Сучасні інформаційні системи = Advanced Information Systems*. – 2023. – Т. 7, № 3. – С. 54-59.
- [11] *Chalyi Serhii* Можливісна модель каузального зв'язку по вхідній змінній для побудови пояснення в інтелектуальній системі / Serhii Chalyi, Volodymyr Leshchynskiy // *Системи управління, навігації та зв'язку. Збірник наукових праць*. – Полтава: ПНТУ, 2023. – Т. 3 (73). – С. 138-143. – doi:<https://doi.org/10.26906/SUNZ.2023.3.138>.
- [12] *Akula, Arjun R., et al.* "CX-ToM: Counterfactual Explanations with Theory-of-Mind for Enhancing Human Trust in Image Recognition Models" // *arXiv preprint arXiv: 2109.01401*. – 2021. (accepted to *iScience* 2021).
- [13] *A. Akula, S. Wang, and S.-C. Zhu,* "CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines", *AAAI*, vol. 34, no. 03, pp. 2594-2601, Apr. 2020.
- [14] *Dubois, Didier and Prade, Henri,* Possibility Theory, Probability Theory and Multiple-valued Logics: A Clarification // *Annals of Mathematics and Artificial Intelligence*. – 2002. – Vol. 32, pp.35–66.

Надійшла до редколегії 14.11.2022

UDC

DOI 10.30837/bi.2022.1(98).12

Yevhen Kupriianov¹¹NTU “KhPI”, Kharkiv, Ukraine, Ukraine, eugeniokupriianov@gmail.com,

ORCID iD: 0000-0002-0801-1789

THEORY OF L-SYSTEMS AS A FRAMEWORK FOR CREATING INFLECTIONAL E-DICTIONARIES

One of the main issues in modern e-lexicography is the creation and support of computer resources meant for teaching language grammar. Among them are electronic dictionaries of word inflection. The compilation of such dictionaries requires not only the choice of computer technologies, but also an appropriate theoretical basis for building a conceptual model that would formally represent the inflectional system of a particular language and serve as a basis for developing a respective dictionary database and interface. For the purposes of this research, such a basis is the theory of L-systems by the leading Ukrainian Academician Volodymyr A. Shyrovok. The author of the paper shares his own experience of using L-systems theory to work out a methodology and implement on its basis a number of inflectional e-dictionaries projects covering Spanish word inflection, English verb inflection, and noun inflection of the Inter-Slavic language. The theory of L-systems can be considered universal, so it is applicable to any other natural or even artificial language with a developed word inflection system. The dictionaries created using this methodology allow: 1) automatic generation of inflectional paradigm for any headword; 2) grouping the words into types, groups, and classes based on the similarity of their inflectional properties; and 3) indicating morphological characteristics of any word form composing a paradigm.

COMPUTER LEXICOGRAPHY, FORMAL MODEL, L-SYSTEM, INFLECTIONAL E-DICTIONARY,
DIGITAL ENVIRONMENT

Купріянов Є. Теорія Л-систем як концептуальна основа для створення словозмінних електронних словників. Однією з головних проблем сучасної електронної лексикографії є створення та підтримка комп'ютерних ресурсів, призначених для навчання граматики мови. Серед них – електронні словники відмінювання слів. Укладання таких словників потребує не лише вибору комп'ютерних технологій, а й відповідного теоретичного підґрунтя для побудови концептуальної моделі, яка б формально репрезентувала флективну систему певної мови і слугувала основою для розробки відповідної словникової бази даних та інтерфейсу. У межах цього дослідження такою основою є теорія Л-систем провідного українського академіка Володимира Анатолійовича Широкова. Автор статті ділиться власним досвідом використання теорії Л-систем для розробки методології та реалізації на її основі низки проектів флективних електронних словників, що охоплюють флексію іспанських слів, флексію англійських дієслів та флексію іменників міжслов'янської мови. Теорію Л-систем можна вважати універсальною, тому вона може бути застосована до будь-якої іншої природної або навіть штучної мови з розвинутою системою словозміни. Словники, створені за цією методологією, дають змогу 1) автоматично генерувати флексійну парадигму для будь-якого заголовного слова; 2) об'єднувати слова в типи, групи і класи на основі подібності їхніх флексійних властивостей; 3) вказувати морфологічні характеристики будь-якої словоформи, що входить до складу парадигми.

КОМП'ЮТЕРНА ЛЕКСИКОГРАФІЯ, ФОРМАЛЬНА МОДЕЛЬ, Л-СИСТЕМА, ЕЛЕКТРОННИЙ СЛОВОЗМІННИЙ СЛОВНИК, ЦИФРОВЕ СЕРЕДОВИЩЕ

Introduction

With large amounts of information growing at ever-increasing rates, the problem of developing effective tools for automatic natural language processing remains relevant. The key objects of natural language processing are lexical units, i.e. words. Therefore, an integral component of every NLP system should be a module for morphological analysis. Such a module is mainly a grammar dictionary generating the full set of inflectional forms, or giving its lemma form from any inflectional form for a particular unit. This is especially true for languages with a developed inflection system, such as Ukrainian, English, German, Spanish, and others. These dictionaries should also have provisions to be used not only by NLP systems, but also by language learners. In this regard, it is necessary to develop an appropriate methodology to elaborate a formal model of the inflection system and develop a database and interface on its basis.

The methodology is possible to be elaborated owing to the theory of lexicographic systems (L-systems), which provides a universal conceptual basis for modeling morphological subsystems for inflected languages. The theory offers a framework for building a formal model to represent the structure and functioning of the word inflection system. This model reflects in a formal way the structural elements of the dictionary, as well as the relations between them. When developing a formal model, it is necessary to determine the types of linguistic information to be provided by the e-dictionary. These include: 1) the inflectional paradigm for both a single linguistic unit and a group of units inflected by the same word change rule 2) the morphological characteristics of any word form and the parameters that determine the place of the unit in the word change system of a particular language; 3) the original form of any word taken from the text.

Based on the above, the purpose of our research is to highlight some issues related to the development of a methodology for compiling inflectional e-dictionaries using the theory of lexicographic systems (L-systems). The methodology was applied while carrying out specific projects of e-dictionaries of: 1) Spanish word inflection, 2) English verb inflection, and 3) noun inflection in the Inter-Slavic language. The projects were implemented under the supervision of the author within the research and development works by the Interdepartmental Center for Applied Linguistics at the National Technical University “Kharkiv Polytechnic Institute”.

1. Related Works

The problems of creating inflectional e-dictionaries are widely discussed in many works [1-8]. The range of issues considered mainly includes:

1. Data science and linguistic foundations necessary for the development of inflectional dictionary-making systems.
2. Compiling inflectional e-dictionaries for NLP systems (morphological analyzers, spell-checkers, information extraction systems, etc.).
3. Developing interface schemes for inflectional e-dictionaries to make more interactive the process of grammar learning.
4. Formation of search queries to work with the database of the word dictionary.
5. Designing the entry structure of inflectional e-dictionaries and selecting parameters for describing inflectional paradigm.

So, the elaboration of an electronic inflectional dictionary suggests a wide range of issues to be resolved in complexity, from collection and analysis of linguistic material to the development of a formal model for representing the inflectional system in the digital environment. The model is used to build up a database and design an interface of e-dictionary. All of this requires an appropriate methodology. Thus, the present research proposes a methodology for compiling word-changing dictionaries that will help solve a number of lexicographic tasks:

- Building a formal model to represent the inflection system in digital environment.
- Parameterization of the inflection system represented by the model.
- Formation of the dictionary database.
- Development of the interface offering different modes of displaying the inflection system.
- Elaboration of an algorithm for the automatic construction of inflectional paradigm for any word.

2. Methodology

The proposed methodology of e-dictionary compilation is based on the theory of L-systems by the leading Ukrainian academician Volodymyr A. Shyrovok [6-8]. As an example, there have been chosen inflection system of English, Spanish, and inter-Slavic languages.

2.1. General overview

An L-system is a specific type of information system that is similar in its structure and functioning to any natural language. The structure of L-system has something like a dictionary and a grammar, which interact with each other to form one or more lexicographic effects (L-effects). Such an effect can be considered any linguistic phenomenon that takes place during language functioning, for example: word formation, lexical groups, synonymy, homonymy, polysemy, etc. In this research, such a lexicographic effect is the generation of inflectional paradigm, as well as the distribution of linguistic units by inflectional types, groups and classes.

2.2. System triad

According to the L-system theory, a language system can be represented by the symbolic triad “structure-substance-subject” or

$$S = S + S + S, \quad (1)$$

where S on the left side stands for the concept of the system, and the right side of the equation demonstrates the interaction of the constituent components of this concept (actually: Structure, Subject, and Substance).

During its functioning, a system shows structural and substantive properties. The structural ones characterize how many initial elements are used to form a particular linguistic object, what the order of these elements should be, and what relations they are combined within the constructed object. The structure is presented as a certain abstract model on which the linguistic material (substance) is overlaid, and as a result we have a concrete linguistic object in the form of a word form, phrase, sentence, etc. At the same time, Volodymyr A. Shyrovok notes that the structure and the choice of appropriate means of materializing it are set by the third element of the triad, the subject, according to the relevant rules and algorithms. In case of inflection:

- the structure is represented by a formal model showing the permanent (“stem”) and variable (“flexion”) parts of a word;
- the substance is material (stems and flexions) to build up corresponding inflectional form of a word;
- the subject applies the algorithms to build an inflectional paradigm using the given structure and substantive elements.

2.3. Inflection phenomenon as a L-effect

The L-effect is a product of language functioning in the form of system triad (1). As for inflection, it can be formally represented as:

$$H : [x] = c(x) * [f(x)], \quad (2)$$

where H denotes L-effect (inflection), $[x]$ is a set of word forms (paradigm) of unit x , and $[f(x)]$ is a set of quasi-flexions that are part of the paradigm $[x]$. The symbol “*” denotes the concatenation of a quasi-stem with a quasi-flexion. The

prefix quasi- means that the terms “stem” and “flexion” do not always correspond to their definitions in traditional linguistics.

Thus, analyzing how conjugation is performed, we can see that quasi-flexions can cover the whole word (in the case of the suppletive forms), part of the stem, or coincide with the ending. For suppletive forms, we have a zero quasi-stem. Any unit x is supposed to have a corresponding quasi-stem $c(x)$ and a set of corresponding quasi-flexions $f(x)$. Table 1 shows the examples of the elements in three languages.

Table 1
Quasi-stem and quasi-flexions in three languages

English		Spanish		InterSlavic	
$c(x)$	$f(x)$	$c(x)$	$f(x)$	$c(x)$	$f(x)$
ha	ve	t	engo	ljub	ov
ha	s	t	ienes	ljub	ve
ha	s	t	iene	ljub	vi
ha	ve	t	enemos	ljub	ov
ha	ve	t	enéis	ljub	ovju
ha	ve	t	ienen	ljub	vi

Table 1 shows an example of quasi-stems and quasi-flexions used to form the paradigm of the Spanish verb *tener* (*to have*), the English verb *to have* and the Inter-Slavic noun *ljubov* (*love*). The formal model (2) shows, at a higher abstract level, the mechanism of building a paradigm for a word regardless the part of speech it belongs to: noun, adjective or verb, etc. It is important to note that the quasi-stem $c(x)$ and the quasi-flexions $f(x)$ can take on the following values:

- $c(x) = \emptyset$ if the word has suppletive forms, such as the English verb *to be*, the Spanish verb *ser* (*to be*), and the inter-Slavic noun *člověk* (*man*);
- $f(x) = \emptyset$ if the word is uninflected, for example, the English verb *must*, all Spanish numerals, adverbs, and some nouns, as well as indeclinable nouns of inter-Slavic languages such as *taksi*, *žuri*, *depo*;
- $c(x) = x - f(x)$ if the word in its formal structure has both the quasi-stem and respective quasi-flexions, e.g. English verb *to steal*, Spanish verb *tomar* (*to take*) or the Inter-Slavic noun *žena* (*wife*).

The set of quasi-flexions $[f(x)]$ depends on the word type. Each of them is divided into grammatical classes and groups and paradigmatic classes.

2.4. Inflectional classification principle

The theory of L-systems provides the following distinguishing features for the universal classification of linguistic units: paradigmatic type, grammatical class and paradigmatic classes. Paradigmatic type is a set of language units with the same grammatical function, which are inflected by the same sets of inflection parameters. Each paradigmatic type can cover several grammatical classes, i.e. a set of words united by common grammatical features.

In some languages, grammar classes can be also divided into paradigmatic groups – groups of words that represent a certain type of inflection paradigm (for example, regular or irregular paradigm, or even double paradigm for verbs). A paradigmatic group consists of paradigmatic classes – a set of words that in the process of forming an inflectional paradigm use the same set of endings (quasi-flexions).

Paradigmatic types will be denoted by T_i , $i = 1, 2, \dots, N$, where N is the number of paradigmatic types; $W(T_i)$ is the set of words belonging to type T_i . Then $K = \{K_1, K_2, \dots, K_n\}$ is the set of grammatical categories of a language, and $\Omega(T_i)$ is the set of grammatical values corresponding to a certain paradigmatic type T_i . Thus, the multiset of the words comprising the paradigmatic type can be given in the following formal form:

$$\langle x|TK \rangle = [\langle x, K_1 \rangle \delta(x, K_1); \dots, \langle x, K_n \rangle \delta(x, K_n)], \quad (3)$$

where $\delta(x, K_i)$ denotes the factor of occurrence of a word x to the paradigmatic type provided it has grammatical feature K_i . This factor can be defined in the following way:

$$\delta(x, K_i) = \begin{cases} 1, & \text{if } x \text{ has a } K_i \text{ appropriate for given } T, \\ 0, & \text{if } x \text{ has a } K_i \text{ in appropriate for given } T, \end{cases} \quad (4)$$

Each paradigmatic type is characterized by its own set of features expressed through grammatical categories, grammatical meanings, and forms. The number of paradigmatic types may vary in a given language: T_i , $i = 1, 2, \dots, N$, where N is the number of paradigmatic types. So, all the set of the words W in any language can be grouped into several paradigmatic types:

$$W = W(T_1) \cup \dots \cup W(T_n), \quad (5)$$

where $W(T_1)$ and $W(T_n)$ are the sets of the words corresponding to a particular paradigmatic type T by their features.

Further, the paradigmatic type can be divided into several grammatical classes, i.e., sets of words W^{T_i} the structure of which can change as a result of applying a particular inflectional parameter. The subdivision of words into grammatical classes within a certain paradigmatic type can be described as follows:

$$W^{T_i} = \bigcup_{j=1}^p W(P_j), \quad (6)$$

where P_j , $j = 1, 2, \dots, p$ is grammatical class, and p is the number of grammatical classes in the paradigmatic type T .

A grammatical class may comprise one or several paradigmatic classes, which can be represented as $\Pi_k \subseteq P_j \subseteq T_i$. This expression should be read as follows: inside the paradigmatic class Π_k there are words of the grammatical class P_j , whose inflection belongs to the paradigmatic type T_i . Thus, the set of the words with an inflectional paradigm of type T_i takes the final form:

$$W(T_i) = \bigcup_{j=1}^{p_i} \left(\bigcup_{k=1}^{n_j} W(\Pi_k) \right), \quad (7)$$

where p_j is the number of grammatical classes in which the inflection is of paradigmatic type T_i ; n_j is the number of paradigmatic classes of grammatical class P_j in which the inflection is of paradigmatic type T_i .

The general principle of classifying the words by their inflectional properties is shown in Fig. 1.

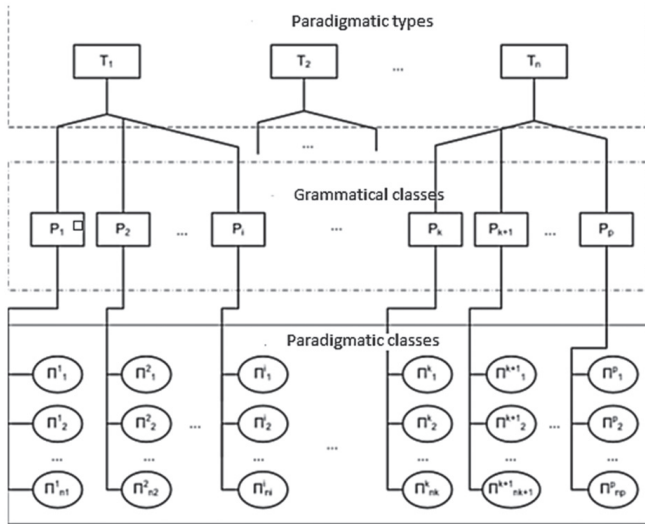


Fig. 1. General principle of grouping the words by inflectional properties

The application of inflectional classification principle to the words of the three languages will be presented in the following subsections.

2.5. L-system model and architecture

This subsection sets out the principle of modeling the inflectional system of any language. Formally, it can be represented as a L-system where the lexicographic effect (inflection) is produced:

$$LS = \{ I^Q(D), \Lambda^\Lambda(I^Q(D)), P^\Lambda(I^Q(D)), F', C', H' \}, \quad (8)$$

where where LS is L-system; $\Lambda^\Lambda(I^Q(D)) = \{x_0\}$ is the set of units in lemmatical form, $P^\Lambda(I^Q(D)) = \{[x]\}$ is the set of inflectional paradigms, F' is an operator establishing the relation “unit – lemmatical form”, C' is an operator establishing the relation “unit – paradigm”, H' is an operator that correlates the lemmatical form with a paradigm. In turn, the elements of L-system $\Lambda^\Lambda(I^Q(D))$ and $P^\Lambda(I^Q(D))$: can be decomposed using the recursive reduction mechanism $RR \downarrow [V(I^Q(D))]$, as shown in Fig. 2.

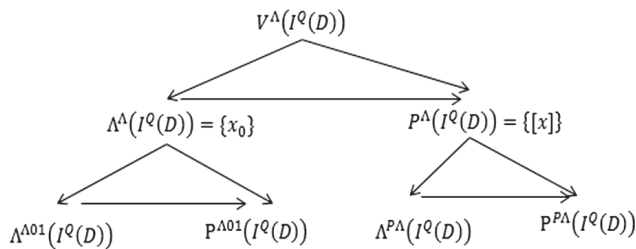


Fig. 2. Recursive reduction $RR \downarrow [V(I^Q(D))]$

The left part $\Lambda^\Lambda(I^Q(D))$ of L-system includes a set of parameters $\Lambda^{\Lambda01}(I^Q(D))$ that determine the place

of a unit in the inflection system according to the classification (paradigmatic type, grammatical class, paradigmatic class), as well as $P^{\Lambda01}(I^Q(D))$ containing the units in lemmatical form that correspond to the parameters $\Lambda^{\Lambda01}(I^Q(D))$. The right part $P^\Lambda(I^Q(D))$: contains both all the word forms that make up the paradigm $P^{P\Lambda}(I^Q(D))$ and the parameters representing the set of grammatical values $\Lambda^{P\Lambda}(I^Q(D)) \equiv \Omega$.

The implementation of a lexicographic system is a type of information system, the architecture of which is chosen to be three-level, compliant with ANSI/X3/SPARK (or simply ANSI/SPARK). The main components of the ANSI/SPARK architecture will be used in this interpretation:

$$ARCH_LS = \{CM, INM, EXM; \Theta, \Psi, \Xi\} \quad (9)$$

where CM denotes conceptual model, $INM = \{InM\}$ is a set of internal models and $EXM = \{exM\}$ represents the set of external models. In case of e-dictionaries CM represents the formal structure of inflectional system, inM is a model showing the e-dictionary database structure and exM designates an interface scheme to access the e-dictionary database.

The application of L-system model (8) when compiling inflectional e-dictionaries will be demonstrated in the following section.

3. Experiment

The experimental part of the research proposed includes the building of a conceptual model in accordance with the methodology outlined, selecting model parameters, as well as designing a database and interface schemes for inflectional e-dictionaries in question.

3.1. E-dictionary of English verb inflection

The first step in compiling the dictionary was to classify English verbs according to their inflectional properties, as described in 3.4. Such classification requires selecting formal criteria by which the whole set of verbs is divided into certain subsets, the mutual intersection of which is empty and within each of them the inflection generation follows the same rule. The subsets of verbs with such properties will be referred to as grammatical classes (6). Each of grammatical classes falls into several paradigmatic groups (7), representing certain types of word change. For example, in English, we distinguish three grammatical classes: “Regular verbs”, “Irregular verbs” and “Defective verbs”. In turn, the second grammatical class can be divided into paradigmatic groups in which inflection is built according to certain rules, for example, doubling or alternating the final consonant, etc. For example, the grammatical class “Irregular verbs” is divided into the following paradigmatic groups: 1) “Verbs with individual inflection in past tense”; 2) “Verbs with an unchangeable past form”; 3) “Verbs with 2 or 3 identical past forms”. In this dictionary the whole inflection system is represented by the single paradigmatic type of verbs. The full diagram showing the English inflectional system of the verbs is shown in Figure 3. The most

developed is the grammatical group "Irregular verbs", as it contains 35 paradigmatic classes, which are arranged in the respective paradigmatic groups.

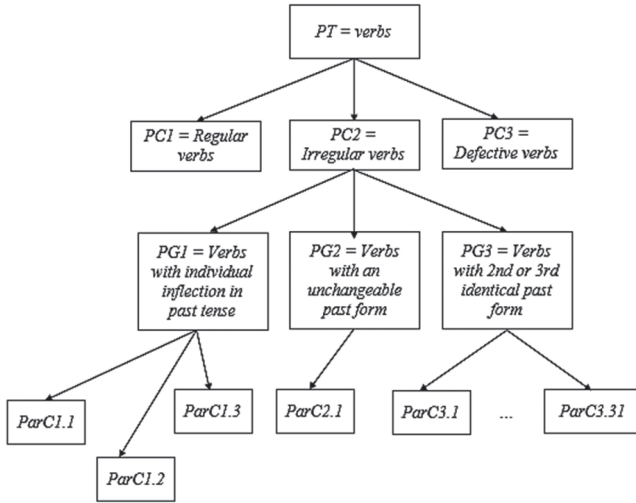


Fig. 3. English inflectional system of the verbs

The above inflectional classification was used on the second step while constructing the conceptual model and its parameterization. The L-system (8) with its structural elements shown in figure 2 has been taken as a basis. The left part of L-system $\Lambda^{\wedge}(I^Q(D))$ represents the inflectional system as a whole and comprises:

- $\Lambda^{\wedge 01}(I^Q(D))$, a set of classification parameters:
 $\Lambda^{\wedge 01}(I^Q(D)) = \{GC, PG, ParC\}$
- $P^{\wedge 01}(I^Q(D))$, a set of English verbs in lemmatical (infinitive) form:

$$I^Q(D) = \{x_0\};$$

The right part $P^{\wedge}(I^Q(D))$: contains various sets of lexicographic descriptions of the English verb paradigm, namely:

- $P^{\wedge \Lambda}(I^Q(D))$, a set of word forms of the verbs as a result of L-effect H in the L-system:
 $P^{\wedge \Lambda}(I^Q(D)) = \{[x]\};$
- $\Lambda^{\wedge \Lambda}(I^Q(D)) \equiv \Omega$, the set of grammatical meanings that a particular form of a verb takes on.

The set of grammatical meanings Ω is composed of several subsets $\Omega = \Omega^1, \Omega^2, \Omega^3, \Omega^4, \Omega^5, \Omega^6, \Omega^7, \Omega^8$, each one having particular parameters:

- $\Omega^1 = \{\omega_1^1, \omega_2^1\} = \{finite, non - finite\}$, the subset of grammatical meanings to characterize the verb forms;
- $\Omega^2 = \{\omega_1^2, \omega_2^2, \omega_3^2\} = \{1st, 2nd, 3rd\}$, the subset of grammatical meanings of person;
- $\Omega^3 = \{\omega_1^3, \omega_2^3\} = \{singular, plural\}$, the subset of grammatical meanings of number;
- $\Omega^4 = \{\omega_1^4, \omega_2^4, \omega_3^4\} = \{present, past, future\}$, the subset of grammatical meanings of tense;
- $\Omega^5 = \{\omega_1^5, \omega_2^5, \omega_3^5, \omega_4^5\} = \{simple, progressive, perfect, perfect progressive\}$, the subset of grammatical meanings of tense type;

- $\Omega^6 = \{\omega_1^6, \omega_2^6, \omega_3^6\} = \{indicative, subjunctive, imperative\}$, the subset of grammatical meanings of mood.
- $\Omega^7 = \{\omega_1^7, \omega_2^7\} = \{active, passive\}$, the subset of grammatical meanings of voice.
- $\Omega^8 = \{\omega_1^8, \omega_2^8, \omega_3^8\} = \{infinitive, gerund, participle\}$, the subset of non-finite forms of a verb.

The next step after building the conceptual model was to develop a database for an e-dictionary of English verb inflection. The database was created using SQLite Studio, a compact built-in database management system. As seen in Fig. 4, the database has six tables: Verbs, Words, Groups, Types, Modal verbs, and Modal groups.

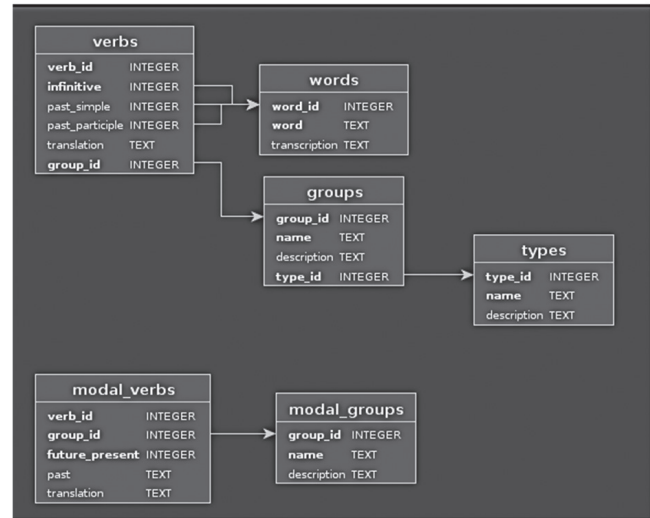


Fig. 4. Database structure

These tables contain linguistic information and additional data for automatic generation of inflectional paradigm and its representation in a dictionary entry. The tables are described briefly below:

- Table "Groups": information about all paradigmatic groups and their relation with respective paradigmatic classes, including id number and short description;
- Table "Words": list of all the English verbs with their unique number, number of paradigmatic group and transcription;
- Table "Verbs": quasi-flexions for the verbs listed in the Table "Words" together with the unique number of paradigmatic group and translation of the verbs;
- Table "Types": description of grammatical classes: regular or irregular verbs;
- Table "Modal_verbs": information related to modal verbs, namely: paradigmatic class numbers and grammar forms.

For automatic generation of the inflectional paradigm, the database contains the information about the quasi-system for any verb. The generation process by the e-dictionary depends on the value of $c(x)$ for a word form, as shown in Table 2.

Table 2

Determination of quasi-stem $c(x)$ to build inflectional paradigm

x	$c(x)$	$c(x)$ value	$f(x)$ value	Fragment of inflectional paradigm
teach	$c(x) = x - f(x)$	t	each	taught, taught
call	$c(x) = x + f(x)$	$c(x)$	0	called
go	$c(x) = \emptyset$	0	went	went, gone

The final step was to develop the dictionary’s interface, which consists of the following elements (Fig. 5): a search bar and a dictionary entry panel. The search bar offers three modes of working with English verb inflection. The first one allows building a paradigm for a specific verb at the user’s request. The second (search by class) and third (search by group) modes display a list of verbs that have common verb-inflectional properties according to the classification performed (Fig. 3).

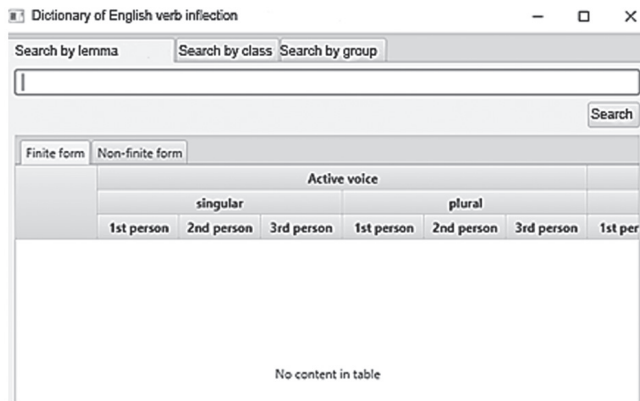


Fig. 5. General view of the dictionary interface

The dictionary entry panel has a certain structure for displaying lexicographic information in accordance with the selected parameters of the conceptual model. The panel is divided into two parts: “Finite form” and “Non-finite form”. The first part displays all the person forms of the verb in the active and passive voice and in all tense and mood categories. Non-Finite form tab displays the participle, infinitive and gerund form of the verb in question.

The “Search by Class” mode offers a list of verbs belonging to a particular grammatical class, including: Irregular verbs, Regular verbs, and Modal verbs. For example, Figure 6 shows an example of displaying all verbs in the Irregular Verbs class.

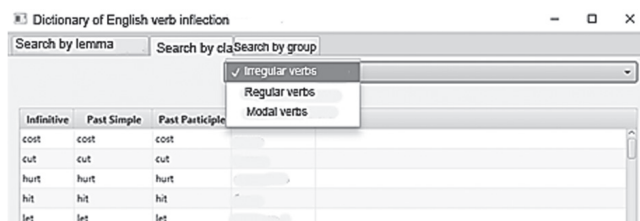


Fig. 6. Search of the verbs by grammatical class

To see verbs categorized by paradigmatic groups, navigate to the “Search by Group” page and choose the desired group from the list. The paradigmatic group named “Irregular verbs – group 1” is shown in Fig. 7.

The developed dictionary of English verb inflection is intended to be used in linguistic research and educational process.

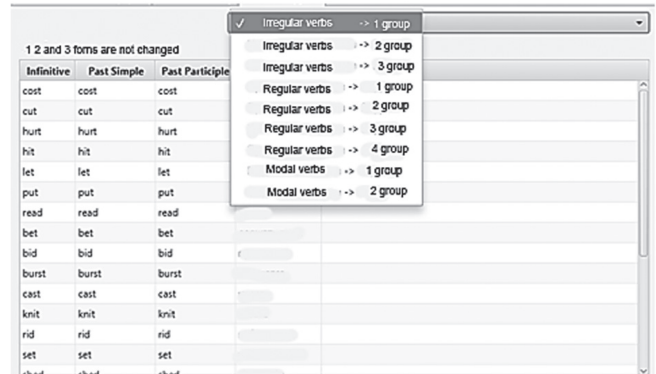


Fig. 7. Example of paradigmatic group 1 of irregular verbs

3.2. E-dictionary of noun inflection in the Inter-Slavic language

Inter-Slavic is a semi-artificial language used by Slavs of different nationalities to communicate with each other. When developing the dictionary, we used the data available on the website of the author of the Inter-Slavic language and performed certain operations to achieve the look we needed. The dictionary currently has about 8000 words.

The first task in creating a dictionary is to build the inflectional classification of inter-Slavic nouns. As in most Slavic languages, nouns in Inter-Slavic have three genders (masculine, feminine, neuter), two numbers (singular, plural), and seven cases (nominative, genitive, dative, accusative, instrumental, local, and vocative). Since only nouns are considered in this project, there is only one paradigmatic type. This type is divided into 2 grammatical classes: inflected nouns (GC1) and uninflected nouns (GC2). The first class contains 3 paradigmatic groups: nouns with a complete paradigm (PG1), nouns with a defective singularia tantum paradigm (PG2), and nouns with a defective pluralia tantum paradigm (PG3). Each paradigmatic class has 14 grammatical forms in the full paradigm. For example, the word *posol* (ambassador) falls into the paradigmatic class of the words with the following set of quasi-flexions {ol, la, la, lu, lom, lu, le, li, lov, lov, lam, lami, lah, li}. The characteristic of paradigmatic groups is given in Table 3. and the classification scheme of Interslavic nouns is shown in Fig. 8.

Table 3

Group name	Numb. of paradigmatic classes
Full paradigm	182
Defective paradigm (singularia tantum)	94
Defective paradigm (pluralia tantum)	35

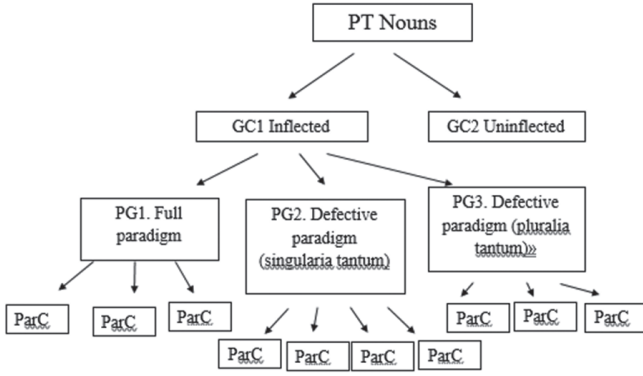


Fig. 8. The classification scheme of Interslavic noun

The L-system (8) for the dictionary was built in the same way as in 3.1. The left part of L-system $\Lambda^{\wedge}(I^Q(D))$ represents the inflectional system of the nouns and includes:

- $\Lambda^{\wedge 01}(I^Q(D))$, a set of classification parameters:

$$\Lambda^{\wedge 01}(I^Q(D)) = \{GC, PG, ParC\};$$
- $P^{\wedge 01}(I^Q(D))$, a set of the nouns in lemmatical form:

$$I^Q(D) = \{x_0\};$$

The right part $P^{\wedge}(I^Q(D))$: consists of the sets of lexicographic descriptions of the noun paradigm, namely:

- $P^{\wedge A}(I^Q(D))$, a set of word forms of the nouns as a result of L-effect H in the L-system:

$$P^{\wedge A}(I^Q(D)) = \{[x]\};$$
- $\Lambda^{\wedge PA}(I^Q(D)) = \Omega$, the set of grammatical meanings that a particular form of a noun takes on.

The set of grammatical meanings Ω is composed of two subsets $\Omega = \{\Omega^1, \Omega^2\}$, each one having particular parameters:

$$\Omega^1 = \{\omega_1^1, \omega_2^1, \omega_3^1, \omega_4^1\} =$$

$$\left\{ \begin{array}{l} \textit{singular, plural,} \\ \textit{singularia tantum,} \\ \textit{pluralia tantum} \end{array} \right\},$$

the subset of grammatical meanings of number;

$$\Omega^2 = \{\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2, \omega_5^2, \omega_6^2, \omega_7^2\} =$$

$$\left\{ \begin{array}{l} \textit{nominative, genitive,} \\ \textit{dative, accusative,} \\ \textit{local, vocative} \end{array} \right\},$$

the subset of grammatical meanings of case.

The database of the dictionary has almost similar tables as the dictionary of the verb inflection. The interface offers three modes of representing of noun inflection as it shown in Figures 9-11.

Given that the dictionary is written for Latin words, the automatic noun declension system of the inter-Slavic language will also accept Latin words as input.

	singular	plural
Nom.	uho	uši
Acc.	uho	uši
Gen.	uha	ušij
Dat.	uhu	ušam

Fig. 9. Single word mode

	alimenty	Balkany	Blizneci	blizneci
	plural	plural	plural	
Nom.	alimenty	Balkany	Blizneci	Nom.
Acc.	alimenty	Balkany	Blizneci	Acc.
Gen.	alimient	Balkan	Bliznecij	Gen.
Dat.	alimientam	Balkanam	Bliznecam	Dat.
Inst.	alimientami	Balkanami	Bliznecami	Inst.

Fig. 10. Grammatical group mode

	alimenty	Balkany	čary	čtkody
	plural	plural	plural	
Nom.	alimenty	Balkany	čary	Nom.
Acc.	alimenty	Balkany	čary	Acc.
Gen.	alimient	Balkan	čary	Gen.
Dat.	alimientam	Balkanam	čaram	Dat.
Inst.	alimientami	Balkanami	čarami	Inst.
Loc.	alimientah	Balkanah	čarah	Loc.

Fig. 11. Paradigmatic class mode

3.3. E-dictionary of Spanish inflection

The following paradigmatic types are distinguished in the Spanish word inflection system: nouns and adjectives T_1 , verbs T_2 , personal and reflexive pronouns T_3 , articles T_4 , and T_0 which includes uninflected words in Spanish (adverbs, prepositions, conjunctions, interjections). Each of the identified paradigmatic types is characterized by a certain set of inflection parameters (see Table 4).

The set of quasi-flexions is determined by the unit's relation to the relevant paradigmatic class. In modern Spanish, we have identified 239 paradigmatic classes, which are distributed in 8 lexico-grammatical classes, 25 paradigmatic groups within 6 paradigmatic types.

A linguistic unit has a set of quasi-flexions corresponding to the paradigmatic type, grammatical class, grammatical group of regular words with a full paradigm, and paradigmatic class of verbs of the first conjugation. For example, for the verb *tomar* (*to take*), the set of quasi-flexions will look like this: $[f(x)] = \{-o, -as, -a, -amos, -áis, -an, -é, -aste, -ó, -amos, -asteis, -aron, -aba, -abas, -aba, -ábamos, -abais, -aban, -aré, -arás, -ará, -aremos, -aréis, -arán, -ía, -ías, -ía, -íamos, -íais, -ían, -e, -es, -e, -emos, -éis, -en, -ara o -ase, -aras o -ases, -ara o -ase, -áramos o -ásemos, -arais o -aseis, -aran o -asen, -are, -ares, -are, -áremos, -areis, -aren, -a / -á, -e, -ad, -en, -ar, -ando, -ado\}$.

Table 4

Inflection parameters in Spanish

PT	Parameter name	Parameter values
T ₁	number	singular, plural
T ₂	tense	present, past imperfect, past perfect, future
	mode	indicative, conditional, imperative, subjunctive
	person	first, second, third
	number	singular, plural
T ₃	case	nominative, accusative, dative, locative
T ₄	number	singular, plural

The components of the set of grammatical meanings of the Spanish language $\Omega = \{\Omega^1, \Omega^2, \Omega^3, \Omega^4, \Omega^5, \Omega^6, \Omega^7, \Omega^8, \Omega^9\}$ are:

- $\Omega^1 = \{\omega_1^1, \omega_2^1, \omega_3^1, \omega_4^1\} \equiv \{m., f., m. y f., m. o f.\}$: the subset of grammatical meanings of gender;
- $\Omega^2 = \{\omega_1^2, \omega_2^2, \omega_3^2\} \equiv \{singular, plural\}$: the subset of grammatical meanings of number;
- $\Omega^3 = \{\omega_1^3, \omega_2^3, \omega_3^3, \omega_4^3\} \equiv \{\emptyset, presente, preterito perfecto simple, preterito imperfecto, futuro simple\}$: the subset of grammatical meanings of tense;
- $\Omega^4 = \{\omega_1^4, \omega_2^4, \omega_3^4, \omega_4^4\} \equiv \{1a pers., 2a pers., 2a pers.-voseo, 3a pers.\}$: the subset of grammatical meanings of person;
- $\Omega^5 = \{\omega_1^5, \omega_2^5, \omega_3^5, \omega_4^5\} \equiv \{indicativo, condicional, subjuntivo, imperativo\}$: the subset of grammatical meanings of mood;
- $\Omega^6 = \{\omega_1^6, \omega_2^6, \omega_3^6, \omega_4^6\} \equiv \{nominativo, dativo, acusativo, preposicional\}$: the subset of grammatical meanings of case;
- $\Omega^7 = infinitivo, \Omega^8 = gerundio, \Omega^9 = participio$: non-finite forms of the verb.

The structure of the Spanish inflection dictionary database (Fig. 11) is built on the basis of a conceptual model that includes the following tables:

- Table “nom”, which represents the word list together with the code of the grammatical class part and the number of the paradigmatic class (type field);

- Table “indent” that specifies parameters and characteristics that are the same for a particular paradigmatic class;
- Table “flex” containing sets of quasi-flexions;
- Table “Parts” of grammatical classes and their codes;
- Table “gr” of inflectional classes;
- Table “procInPar” of typical sets of typProc procedures for creating analytical forms.

The software interface of the dictionary is currently under development, but its functionality will be similar to that described in subsections 3.1 and 3.2. The main functions will include:

- viewing the word list;
- generating the full word paradigm of the word selected from the wordlist and its main grammatical characteristics;
- displaying a part of the word list (by part of speech, by paradigmatic class number, by an arbitrary query in SQL);
- displaying all grammatical homonyms, proper names, etc.;
- displaying quantitative characteristics regarding the content of paradigmatic classes, parts of speech, homonyms, etc.;
- search for words in the word list;
- building a direct or inverse dictionary (setting direct or inverse sorting in the register);

Lemmatization and morphological analysis of any word or word form.

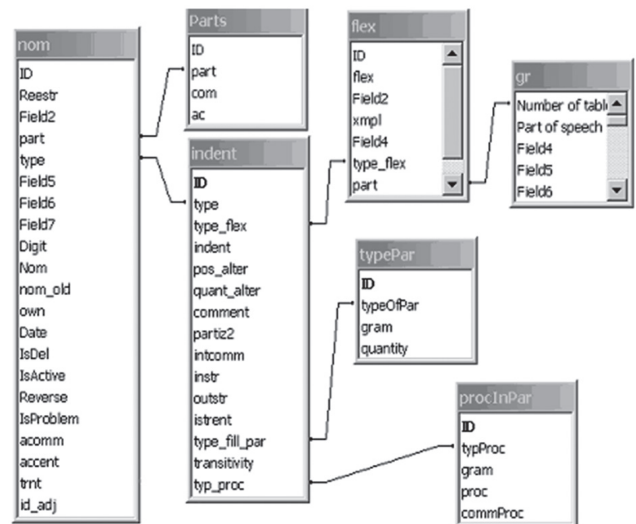


Fig. 12. Spanish inflection dictionary database structure

4. Results

The inflectional e-dictionaries, the compilation of which is described in subclauses 3.1 – 3.3, can be used in different modes:

- single word: its word-shifting paradigm together with the grammatical characteristics of each word form;

- grammatical classes: a list of all words that are grouped together by their inflectional characteristics, along with their paradigm;
- paradigmatic groups: a list of all the words included in the selected paradigmatic group, which is part of a certain grammatical class;

With regard to the use of dictionaries by researchers, it is possible to study the pattern of changes that occur during the formation of a paradigm; the influence of lexical meaning on the realization of word-paradigm units; the phenomena of homography, i.e., units that coincide with each other in their grammatical form but belong to different paradigmatic classes.

For educational purposes, the dictionary can: provide a paradigm for a particular word, provide a list of words that are declined according to the same rules, indicate differences in the word change of words belonging to the same grammatical class or group. The developed models and structure of the e-dictionary make it possible to use it in programs for automatic generation of a complete word change paradigm for any register unit of the dictionary (generation of all possible variants of grammatical forms).

Conclusions

The inflectional e-dictionaries created using the methodology described in Section 3 can be used as:

- automatic morphological analysis tool in or NLP systems (at the stages of POS-tagging, lemmatization and synthesis);
- reference tool (word search, providing information on word pronunciation of specific word units of the dictionary).

Such peculiarities of the dictionary's purpose also put forward certain requirements to its structure (the linguistic information presented in the dictionary should be sufficient to fulfill all the required functions; a wide variety of ways of access to this information should be provided).

References

- [1] *Ducassé M., Elizbarashvili A.* Finding lemmas in agglutinative and inflectional language dictionaries with logical information systems // Proceedings of the XX EURALEX International Congress "Dictionaries and Society". – 2022. – P. 381–386.
- [2] *Marjanović S.* The SERBOVERB language resource and its multifunctionality // Proceedings of the eLex 2023 conference "Electronic lexicography in the 21st century". – 2023. – P. 567–586.
- [3] *Czerepowicka M.* The structure of a dictionary entry and grammatical properties of multi-word units, // Proceedings of the eLex 2023 conference. – 2021. – P. 200–215.
- [4] *Starko V., Rysin A.* VESUM: A large morphological dictionary of Ukrainian as a dynamic tool // Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2022). – Vol. 1: Main conference. – 2022. – P. 61–70.
- [5] *Kluza K.* Using an inflection dictionary for a design of a visual dictionary of verbs and their attributes // Studies and Materials in Applied Computer Science. – 2011. – №3(5). – P. 25–30.
- [6] *Shyrovkov V.* Information, Language, System. – Kyiv: Akadempriodyka, 2021. – 164 p.
- [7] *Liubchenko T. P., Shevchenko I. V., Shyrovkov K. V.* Grammatical lexicographical systems with the flectional and agglutinative component of inflection // *Movoznavstvo*. – 2013. – № 2-3. – P. 209–220.
- [8] *Shyrovkov V.* Hramatychni systemy. Fenomenolohichni pidkhid. – Kyiv: Naukova Dumka. – 2018. – 312 p.

The article was delivered to editorial staff on the 06.12.2022



Iryna Kyrychenko¹, Yehor Shamrai²

¹ Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
iryna.kyrychenko@nure.ua, ORCID: 0000-0002-7686-6439

² Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
yehor.shamrai@nure.ua

COMPARATIVE ANALYSIS OF URL SHORTENING ALGORITHMS: PERFORMANCE CONSIDERATIONS IN THE APPLICATION DOMAIN OF WEB SEARCH AND INFORMATION RETRIEVAL TECHNOLOGY

This article provides an in-depth overview and comparative analysis of URL shortening algorithms, with a focus on their performance and security. URL shortening is an important tool for creating concise versions of long URLs, particularly in platforms with character limitations. However, the increasing prevalence of cyber threats has raised concerns regarding the security of URL shortening. The paper examines a range of URL shortening algorithms, including Hash-based and Randomized algorithms, and compares their effectiveness based on factors such as URL length, randomness, and resistance to brute-force attacks. In addition, the article explores the security risks involved in URL shortening and suggests strategies for mitigating them. The paper also includes a performance analysis of different URL shortening algorithms, considering the time required to generate a shortened URL and redirect the user to the original URL. Finally, the article discusses the challenges of evaluating URL shortening algorithms and outlines potential avenues for future research in this area.

URL SHORTENING ALGORITHMS, PERFORMANCE EVALUATION, SECURITY CONSIDERATIONS, HASH-BASED ALGORITHMS, PERFORMANCE EVALUATION, RANDOMIZED ALGORITHMS

Кириченко І.В., Шамрай Є.О. Порівняльний аналіз алгоритмів скорочення URL-адрес: Міркування щодо продуктивності в прикладній області веб-пошуку та інформаційно-пошукових технологій. У цій статті представлено детальний огляд і порівняльний аналіз алгоритмів скорочення URL-адрес з акцентом на їхню продуктивність і безпеку. Скорочення URL-адрес є важливим інструментом для створення стислих версій довгих URL-адрес, особливо на платформах з обмеженнями на кількість символів. Однак, зростаюча поширеність кіберзагроз викликає занепокоєння щодо безпеки скорочення URL-адрес. У статті розглядається низка алгоритмів скорочення URL-адрес, включаючи алгоритми на основі хешування та рандомізовані алгоритми, і порівнюється їхня ефективність на основі таких факторів, як довжина URL-адреси, випадковість та стійкість до атак грубої сили. Крім того, в статті досліджуються ризики для безпеки, пов'язані зі скороченням URL-адрес, і пропонуються стратегії для їх зменшення. Стаття також містить аналіз продуктивності різних алгоритмів скорочення URL-адрес, враховуючи час, необхідний для генерації скороченої URL-адреси і перенаправлення користувача на оригінальну URL-адресу. Нарешті, в статті обговорюються проблеми оцінки алгоритмів скорочення URL-адрес і окреслюються потенційні шляхи для майбутніх досліджень у цій галузі.

АЛГОРИТМИ СКОРОЧЕННЯ URL, ОЦІНКА ПРОДУКТИВНОСТІ, МІРКУВАННЯ БЕЗПЕКИ, ХЕШ-АЛГОРИТМИ, ОЦІНКА ПРОДУКТИВНОСТІ, РАНДОМІЗОВАНІ АЛГОРИТМИ

Introduction

URL shortening is a process of converting a long URL to a shorter one, which redirects to the original URL. The shortened URL is used to share links, save characters in social media posts, or make it easier for users to remember a URL. URL shortening services have become popular due to their convenience, but they also have some disadvantages, such as security concerns and performance issues. In this article, we will review and compare various URL shortening algorithms based on their performance and security.

In today's digital age, URLs have become an essential component of our daily lives. They are used to access various resources on the internet, ranging from websites to applications. However, the length of URLs can sometimes be a hindrance, especially when it comes to sharing them on platforms that have character limits, such as Twitter. This is where URL shortening comes into play. URL shortening is the process of taking a long URL and

creating a shorter, more manageable version that redirects to the original URL. This process has become increasingly popular over the years, with numerous URL shortening services available on the internet. However, with the rise of cyber threats, the security of URL shortening has become a major concern. To address this issue, researchers have developed various URL shortening algorithms that prioritize both performance and security. These algorithms aim to create short URLs that are resistant to cyber-attacks while ensuring fast and efficient redirection to the original URL.

This is an overview and comparison of various URL shortening algorithms based on their performance. The paper presents a detailed analysis of the most commonly used URL shortening algorithms, including Hash-based, Bijective, and Randomized algorithms, among others. The analysis considers various factors, such as the length of the generated URLs, the randomness of the generated strings, and the resistance to brute-force attacks. Furthermore,

the paper discusses the security risks associated with URL shortening and the measures that can be taken to mitigate them. It also highlights the importance of using HTTPS [1] protocol for securing the redirection process.

Overall, this paper aims to provide a comprehensive overview of URL shortening algorithms and their associated security risks. It is intended for researchers and practitioners interested in improving the performance and security of URL shortening algorithms.

URL shortening is a process of converting a long URL to a shorter one, which redirects to the original URL. The shortened URL is used to share links, save characters in social media posts, or make it easier for users to remember a URL. URL shortening services have become popular due to their convenience, but they also have some disadvantages, such as security concerns and performance issues. In this article, we will review and compare various URL shortening algorithms based on their performance and security.

In addition to providing an overview and comparison of URL shortening algorithms, this paper also explores the various use cases of URL shortening, such as social media sharing, email marketing campaigns, and tracking user behavior. URL shortening is often used in social media sharing, where the character limit is restricted. Shortened URLs allow users to share links to websites or content that they find interesting or relevant without taking up too much space. In email marketing campaigns, shortened URLs are used to track clicks and user engagement. The ability to track user behavior provides valuable insights into the effectiveness of the campaign. However, despite the numerous benefits of URL shortening, it comes with its fair share of security risks. Attackers can use shortened URLs to launch phishing attacks, spread malware, and perform other malicious activities. Therefore, it is important to consider the security implications of URL shortening algorithms when developing and using them. Finally, the paper discusses the challenges associated with evaluating URL shortening algorithms and the future research directions in this field. It highlights the need for further research to improve the security and performance of URL shortening algorithms while considering the evolving threat landscape.

In summary, this paper provides a comprehensive overview of URL shortening algorithms [2], such as Base62 [3], their associated security risks, SEO [4] optimization and their performance comparison. It is a valuable resource for researchers, developers, and practitioners interested in improving the efficiency and security of URL shortening algorithms.

1. Problematic of URL shortening algorithms and main goals of shortening

URL shortening is a technique that has become increasingly popular in recent years, as it allows users to share long, complicated URLs in a compact and easy-to-read

format. However, the practice of URL shortening also raises a number of concerns and challenges, which need to be addressed in order to ensure that users are able to use shortened URLs in a safe and effective manner.

URL shortening algorithms are used to convert lengthy URLs into shorter ones. This is done to make it easier to share links on social media platforms and other places where character limits are imposed. While URL shortening algorithms have become quite popular, they also come with several problems that need to be addressed.

Security concerns: One of the most significant problems with URL shortening algorithms is the security risks they pose. Because the original URL is obscured by the shortened URL, users may be redirected to malicious or harmful websites without their knowledge. Cybercriminals can use shortened URLs to disguise phishing scams, malware downloads, and other types of online threats. Additionally, shortened URLs can be vulnerable to hacking and tampering, which can compromise user data and privacy.

Link rot: Another problem associated with URL shortening algorithms is link rot, which occurs when the original URL is no longer valid or has been changed. Because shortened URLs rely on the original URL to redirect users to the correct destination, link rot can lead to broken links and frustration for users. While some URL shortening services provide automatic link checking and updating, this is not always the case.

Over-reliance on third-party services: Many URL shortening services are provided by third-party companies, which means that users may have limited control over the service and how it is used. If the service provider goes out of business or changes their policies, this can lead to disruptions in service and broken links. Additionally, third-party URL shortening services may collect user data or display ads, which can be a privacy concern for some users.

Duplication of URLs: Because URL shortening algorithms use a fixed-length code to generate the shortened URL, there is a risk of duplication. If two different URLs generate the same shortened code, this can lead to conflicts and broken links. While some URL shortening services use randomization or other techniques to minimize the risk of duplication, this is not always foolproof.

Dependence on service availability: Finally, URL shortening algorithms are dependent on the availability of the service provider. If the service experiences downtime or other disruptions, users may be unable to access the shortened URLs. Additionally, because URL shortening services are often free or low-cost, there is a risk of service providers going out of business or changing their policies without warning.

By the way, URL shortening comes with several security risks, such as phishing attacks, malware distribution, and the potential exposure of sensitive information. These

risks occur because the shortened URL may obscure the actual destination of the link, and users may not be aware of the actual website they are accessing.

Phishing attacks are a common security risk associated with URL [5] shortening. Attackers may use shortened URLs to redirect users to fake websites that look identical to legitimate ones, tricking users into giving away their sensitive information.

Malware distribution is another significant risk associated with URL shortening. Attackers can use shortened URLs to distribute malware, which can infect the user's device when they click on the link.

To mitigate these risks, several measures can be taken, such as:

1. **HTTPS:** Use HTTPS for all URLs, including shortened URLs, to ensure that all communications between the user's browser and the website are encrypted and secure.
2. **URL scanners:** Use URL scanners that can detect phishing [6] URLs and malware distribution URLs. These scanners can identify and block URLs that are known to be malicious.
3. **User education:** Educate users about the potential risks associated with shortened URLs and how to verify the destination of the link before clicking on it.
4. **URL expiration:** Implement URL expiration policies, which can prevent attackers from using shortened URLs for an extended period.
5. **Randomized URLs:** Use randomized URLs instead of sequential ones. This can make it harder for attackers to guess the destination of the URL.

By taking these measures, the risks associated with URL shortening can be mitigated, and the security of shortened URLs can be improved.

Overall, URL shortening is a useful tool for sharing links on the internet, but it is not without its challenges and limitations. In order to use URL shortening effectively, users need to be aware of the potential problems associated with this practice and take steps to address them, such as monitoring and updating their links regularly, using trusted URL shortening services, and being cautious when clicking on shortened links. By taking these steps, users can minimize the risks associated with URL shortening and enjoy the benefits of this powerful tool for sharing and communicating information online.

2. URL Shortening Algorithms

URL shortening algorithms are used to create shorter and more manageable versions of lengthy URLs. These algorithms take a long URL as input and generate a shorter URL that redirects to the original URL when clicked. URL shortening algorithms are widely used in various applications, including social media, email marketing, and other online platforms that have character limits. One of

the main advantages of using URL shortening algorithms is that they can help save space and improve readability, particularly in social media platforms where character limits are strict. Shorter URLs also make it easier for users to remember and share links and can improve the overall user experience. Another advantage of URL shortening algorithms is that they can provide analytics and tracking capabilities, allowing users to track clicks and monitor the performance of their links. This can be particularly useful for marketing campaigns and other online promotions, as it allows users to track the effectiveness of their efforts and make informed decisions about future campaigns.

3. Hashing Algorithms

Hashing algorithm [7] converts a URL to a fixed-size hash value, which is then used as a short URL. The hash value is unique to each URL, and the algorithm ensures that it produces the same hash value for the same URL. Hashing algorithms are fast and efficient, but they are vulnerable to collisions. A collision occurs when two different URLs produce the same hash value, which can result in a security breach or incorrect redirection.

Hashing algorithms are a popular method used in URL shortening services to generate shortened URLs. Hashing is a mathematical function that takes an input value, such as a URL, and produces a fixed-length output value, known as a hash. Hashing is a one-way function, meaning that it is impossible to reverse the process and obtain the original input value from the hash.

When generating shortened URLs, the hashing algorithm takes the original URL and computes a hash value based on the contents of the URL. The hash value is then mapped to a shorter, more compact URL that redirects to the original URL when clicked. Because the hash value is unique to the original URL, it ensures that each shortened URL is unique and can be used to redirect users to the correct destination.

There are several popular hashing algorithms used in URL shortening services, including MD5, SHA-1, and SHA-256. MD5 is a widely used hashing algorithm that produces a 128-bit hash value, while SHA-256 [8] and SHA-1 are more secure hashing algorithms [9] that produce 160-bit and 256-bit hash values, respectively. The longer the hash value, the more secure the algorithm is, as it becomes more difficult for attackers to guess the original input value.

There are several advantages to using hashing algorithms in URL shortening:

1. **Efficiency:** Hashing algorithms are computationally efficient and can generate hashes quickly and easily. This makes them ideal for use in high-volume applications, such as social media and online advertising.
2. **Uniqueness:** Hashing algorithms produce unique hash values for each input value, which ensures

that each shortened URL is unique and can be used to redirect users to the correct destination.

3. Security: Hashing algorithms are secure [10], as they are one-way functions that make it difficult for attackers to reverse-engineer the original input value from the hash.

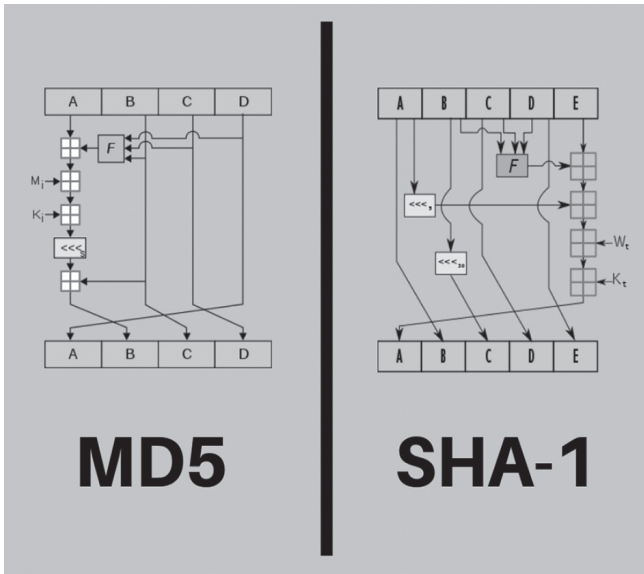


Fig. 1. Block of diagram for MD5 and SHA-1 processors

Random number generation and dictionary-based encoding are popular techniques used in combination with hashing algorithms for URL shortening. Random number generation involves generating a random number or string of characters and using it as a suffix or prefix to the original URL. This can help create unique and shorter URLs, as the length of the original URL no longer determines the length of the shortened URL. However, generating truly random numbers can be challenging, and collisions may still occur if the random numbers are not generated properly. Dictionary-based encoding, on the other hand, involves mapping the original URL to a compressed or encoded version using a predefined set of characters or symbols. This can result in shorter URLs compared to hash-based methods, and collisions are less likely as the encoding scheme is designed to avoid duplicate mappings. However, dictionary-based encoding can also be vulnerable to attacks, such as dictionary attacks or reverse lookups, which can compromise the security of the shortened URLs.

While hashing algorithms are a popular method for generating shortened URLs, they are not without their limitations. One limitation is that the length of the shortened URL is determined by the length of the hash value, which can be longer than other URL shortening methods. Additionally, hash-based URL shortening is vulnerable to collisions, where two different input values produce the same hash value, which can lead to duplicate URLs and other issues. To address these limitations, some URL

shortening services use a combination of hashing and other techniques, such as random number generation or dictionary-based encoding, to create more secure and efficient shortened URLs.

4. Bijective algorithms

Bijective algorithms are commonly used for URL shortening because they are easy to implement, fast, and can create shorter URLs than other types of algorithms. A bijective function is a function that has a one-to-one correspondence between its domain and range, meaning that each input has a unique output, and each output has a unique input. This makes it ideal for generating unique, shortened URLs for long and unwieldy URLs. Functions can be injections (one-to-one functions), surjections (onto functions) or bijections (both one-to-one and onto). Informally, an injection has each output mapped to by at most one input, a surjection includes the entire possible range in the output, and a bijection has both conditions be true.

One of the advantages of bijective algorithms is that they do not require any additional storage space, as the shortened URL can be generated on-the-fly. This makes them particularly useful in situations where storage space is limited, such as in mobile applications or on websites with high traffic volumes.

Another advantage of bijective algorithms is that they can be easily customized to suit specific needs. For example, some applications may require shorter URLs than others, or may need to generate URLs with specific patterns or formats.

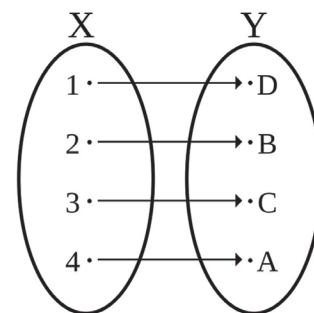


Fig. 2. Bijection between two sets

However, bijective algorithms do have some limitations. One of the main limitations is that the length of the shortened URL is directly proportional to the size of the domain space, which means that the shorter the URL, the smaller the domain space. This can make the algorithm more susceptible to collisions, where two or more URLs map to the same shortened URL. To mitigate this risk, many bijective algorithms use a combination of random numbers and hashes to generate unique, shorter URLs. This can increase the size of the domain space and reduce the likelihood of collisions.

5. Randomized algorithms

Randomized algorithms are algorithms that incorporate some element of randomness or probability into their operation. Instead of producing a deterministic output for a given input, randomized algorithms make use of a random number generator to generate some component of their output. This randomization can be used to improve the efficiency or accuracy of the algorithm, or to add an element of unpredictability or randomness to its behavior. One common application of randomized algorithms is in cryptography and computer security. For example, one of the most widely used cryptographic protocols, the RSA algorithm, relies heavily on random number generation to generate the keys used for encryption and decryption. Similarly, many secure communication protocols, such as SSL/TLS, incorporate randomization in order to prevent eavesdropping or interception of data.

There are several advantages to using randomized algorithms in security applications:

1. **Randomness can make attacks more difficult:** By incorporating randomization into the algorithm, it can be more difficult for an attacker to predict or analyze the behavior of the algorithm. This can make it more difficult for an attacker to find weaknesses or vulnerabilities that can be exploited.
2. **Improved efficiency:** Randomized algorithms can be faster or more efficient than deterministic algorithms, particularly for problems that are difficult to solve using traditional methods. This can be particularly useful for large-scale or computationally intensive security applications.
3. **Unpredictability:** Randomized algorithms can add an element of unpredictability or randomness to the behavior of a system. This can make it more difficult for an attacker to predict the behavior of the system, which can make it more difficult to launch successful attacks.

By the way, there are also some potential disadvantages to using randomized algorithms in security applications:

1. **Potential for bias:** If the random number generator used in the algorithm is biased or predictable, it can be exploited by an attacker to gain an advantage or to break the security of the system.
2. **Difficulty of analysis:** Randomized algorithms can be more difficult to analyze and prove correct than deterministic algorithms, particularly for complex systems. This can make it more difficult to identify or fix vulnerabilities in the system.
3. **Complexity:** Randomized algorithms can be more complex to implement and maintain than deterministic algorithms. This can make them more difficult to deploy or update, particularly in large-scale or distributed systems.

Randomized algorithms can be used for URL shortening by generating a random sequence of characters that

represent the shortened URL. Here are some examples of randomized algorithms that can be used for URL shortening.

A simple approach is to generate a random string of characters to represent the shortened URL. This can be accomplished using a random number generator and a mapping of characters to their corresponding ASCII codes. The resulting string can then be appended to a base URL to create the shortened URL. Bloom filters are commonly used in URL shortening to check if a URL has already been shortened or not. A Bloom filter is a probabilistic data structure that uses multiple hash functions to map a URL to a set of bits in a bit array. When a new URL needs to be shortened, its presence in the Bloom filter is checked by computing the hash values using the same hash functions used to insert the URLs into the filter. If all the corresponding bits in the Bloom filter are set to 1, the URL is considered to be already shortened. If the URL is not already shortened, a randomized algorithm can be used to generate a unique short code for the URL. Bloom filters are widely used in URL shortening services because they are efficient in terms of memory and computation and can provide a high probability of detecting whether a URL has already been shortened. However, Bloom filters have a trade-off between memory usage and false positive rate. If the Bloom filter is too small or uses too few hash functions, the false positive rate can become unacceptably high, resulting in URLs being incorrectly flagged as already shortened. On the other hand, if the Bloom filter is too large or uses too many hash functions, the memory usage and computation time can become prohibitively high. Therefore, it is important to choose appropriate parameters for the Bloom filter based on the expected number of URLs and the desired false positive rate.

6. Comparison of URL shortening algorithms

We are going to cover experimental results of comparison of some type algorithms which can be used for URL shortening. As evaluation and comparison parameters for the algorithms, we will consider their performance and reliability. We will take one representative algorithm from each method of shortening URLs and conduct a detailed analysis, describing the essence of the algorithm and its implementation, testing its performance on different datasets, and comparing each of them. Many of those algorithms are used generally in neural networks and Face detection algorithms.

Let's take a look at SHA-2, Base62 and Bloom Filter algorithms for URL shortening.

6.1. SHA-1

SHA-2 is a family of cryptographic hash functions that were designed by the National Security Agency (NSA) and published by the National Institute of Standards and Technology (NIST) in 2001. The SHA-2 family includes six different hash functions, each with a different output

size: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256. SHA-2 was created as a successor to SHA-1, which had been found to have vulnerabilities that made it insecure. The design of SHA-2 was based on the principles of the Merkle–Damgård construction, which is a popular method for building cryptographic hash functions. The SHA-2 family of hash functions has been widely used in a variety of applications, including digital signatures, password storage, and data integrity verification. It has also been used in conjunction with other cryptographic algorithms, such as RSA and AES, to provide secure communication over the internet.

The hash function is initialized with a set of constants that depend on the specific variant of SHA-2 being used. These constants are known as the initial hash values or initial states, and they are typically fixed for a given variant of the algorithm. For example, the initial hash values for SHA-256 are the first 32 bits of the fractional parts of the square roots of the first eight prime numbers. The input message is padded so that its length is a multiple of 512 bits. The padding is designed to ensure that the message is of a fixed length and to provide additional security against attacks. The padding is done in two steps:

1. Append a single '1' bit to the end of the message.
2. Append '0' bits until the length of the message is congruent to 448 modulo 512. This means that the message length will be a multiple of 512 after the padding, but there will be exactly 64 bits left to store the length of the original message.

The padded message is divided into 512-bit blocks, which are then processed by the hash function. Each block is processed in turn using a set of logical operations that depend on the specific variant of SHA-2 being used. The processing of each block consists of the following steps:

1. The 512-bit block is divided into 16 words of 32 bits each. These words are used to create a message schedule of 64 words, using a specific algorithm that depends on the specific variant of SHA-2 being used.
2. Initialize working variables: The working variables for the hash function are initialized using the initial hash values for the specific variant of SHA-2 being used.
3. Main loop: The message schedule and working variables are used in a series of logical operations to process the block. This loop is repeated 64 times for each block of data.
4. Update working variables: The working variables are updated based on the results of the main loop.

Once all of the blocks have been processed, the output of the hash function is generated. The output size depends on the specific variant of SHA-2 being used. For example, the output size for SHA-256 is 256 bits. The output is generated by concatenating the final hash values for each block, which are typically represented as a set of 32-bit words.

6.2 Base62

Method for encoding data in a format that uses a set of 62 alphanumeric characters, consisting of both uppercase and lowercase letters and digits. This algorithm is commonly used for shortening URLs and generating unique identifiers.

The Base62 algorithm uses a character set of 62 characters, consisting of 26 uppercase letters, 26 lowercase letters, and 10 digits. These characters are usually represented as a string of characters in the order in which they appear in the character set, such as "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz".

Before encoding the data using the Base62 algorithm, it must first be converted to a base-10 number. This is typically done using an existing encoding method, such as UTF-8 or ASCII. Once the data has been converted to base-10, it can be converted to base-62 using the following steps. Divide the base-10 number by 62, and record the remainder. Continue dividing the quotient by 62 and recording the remainders until the quotient is zero.

The resulting sequence of remainders represents the encoded data in base-62.

Each remainder in the sequence represents a numeric value that corresponds to a character in the Base62 character set. To encode the data, each remainder is mapped to its corresponding character in the character set.

The resulting sequence of characters represents the encoded data in the Base62 format. This encoded data can be used as a shortened URL or a unique identifier. The Base62 algorithm involves converting data to a base-10 number, dividing the base-10 number by 62 to obtain a sequence of remainders, mapping the remainders to characters in a character set, and outputting the resulting sequence of characters as the encoded data.

6.3 Bloom filter

The Bloom filter is a space-efficient probabilistic data structure that is useful for answering set membership queries. It was invented by Burton Howard Bloom in 1970. The main idea behind the Bloom filter is to use a bit array and a set of hash functions to determine whether an element is likely to be a member of a set or not. The filter has a fixed size, and elements are added to it by hashing them multiple times and setting the corresponding bits in the array. To query for membership, the same hash functions are applied to the element, and the corresponding bits in the array are checked. If all of the bits are set to 1, then the element is likely to be in the set. If any of the bits are set to 0, then the element is definitely not in the set.

The Bloom filter algorithm requires the definition of two parameters: the size of the filter and the number of hash functions. The filter size, denoted by m , is the number of bits in the filter, and the number of hash functions, denoted by k , determines the number of times the

input data will be hashed. The Bloom filter is initialized as a bit array of size m with all bits set to 0. The bit array is typically represented as an array of Boolean values. Alternatively, a bit vector or a bitmap can be used to represent the filter. To add an element to the Bloom filter, the element is hashed k times using k independent hash functions. Each hash function produces a hash value that is used to set the corresponding bits in the filter to 1. The specific bits to be set are determined by taking the modulo of the hash value with the size of the filter. In other words, the k hash functions produce k indices, and the corresponding bits in the filter are set to 1. To query whether an element is likely to be a member of the set, the element is hashed k times using the same hash functions used to add elements to the filter. The corresponding bits in the filter are checked, and if all of them are set to 1, the element is likely to be a member of the set. If any of the corresponding bits are 0, the element is definitely not a member of the set. One drawback of the Bloom filter algorithm is that it can produce false positives. This occurs when an element is not a member of the set but the corresponding bits in the filter are set to 1 due to collisions with other elements. The probability of a false positive depends on the size of the filter, the number of hash functions, and the number of elements in the set. To reduce the probability of false positives, the size of the filter and the number of hash functions must be carefully chosen based on the expected number of elements in the set. The Bloom filter algorithm is used in a variety of applications, such as web caching, network routers, distributed systems, and spell checkers. It is also used in DNA sequencing to filter out reads that do not match a reference genome. The Bloom filter is a simple and efficient data structure that can save memory and improve performance in many applications.

Table 1

Speed comparison of methods for different length of data

Technology	1 000	10 000	100 000
SHA-2	5 ms	8 ms	11 ms
Base62	2 ms	4 ms	5 ms
Bloom filter	3 ms	5 ms	9 ms

Base62 is an encoding scheme that converts binary data to a string using a character set of 62 characters (usually the 26 lowercase and 26 uppercase letters of the English alphabet, plus the 10 decimal digits). The resulting string is shorter than the original binary data, making it useful for shortening URLs or generating unique identifiers. Base62 encoding and decoding are simple operations that involve basic arithmetic and string manipulation, so they are very fast and have low computational overhead. Encoding and decoding a string using Base62 can be done in a few microseconds on modern CPUs.

SHA-2, on the other hand, is a family of cryptographic hash functions that generate a fixed-size digest of a

given input data. SHA-2 is widely used for data integrity checking and digital signature verification, as well as password hashing and other security applications. SHA-2 is a computationally intensive operation that involves multiple rounds of bitwise operations and nonlinear functions, so it has higher computational overhead than Base62 encoding. The actual performance of SHA-2 depends on the size of the input data, the number of rounds used, and the hardware and software used to perform the computation. In general, SHA-2 can process data at a rate of several megabytes per second on modern CPUs and can take several milliseconds or more to compute a digest for a large input.

Bloom filters have a low memory overhead and can represent very large sets with a small amount of memory, but their accuracy depends on the size of the filter and the number of hash functions used. In terms of string shortening performance, Base62 is likely the fastest option, followed by Bloom filters and then SHA-256. For Base62 is a commonly used encoding scheme for shortening URLs, but it may not be optimal for hashing algorithms. Consider using a longer encoding scheme, such as Base64 or Base58, which provides more entropy and reduces the risk of collisions. Key stretching is a technique that makes brute-force attacks [11] more difficult by slowing down the hashing process. It involves iterative hashing the input with a salt and a fixed number of rounds, which can be adjusted to increase the computational cost of generating the hash. Ensure that the input to the hash function is properly validated to prevent attacks such as buffer overflows or injection attacks. Hash-based message authentication code (HMAC) can be used to add an additional layer of security to the hash function by incorporating a secret key into the calculation. This can prevent attacks such as message tampering or length extension attacks.

The choice of URL shortening algorithm depends on the specific requirements of the application domain. For applications that require speed and compactness, the Base62 encoding algorithm is a good choice. For applications that require security and uniqueness, the SHA-2 hashing algorithm is more suitable. Finally, the Bloom filter algorithm is a good choice for applications that require speed and scalability, but with lower accuracy requirements.

Conclusions

In conclusion, URL shortening algorithms play an important role in the domain of web applications, where shorter URLs can be beneficial for various reasons, such as improving usability, reducing the risk of errors, and tracking user behavior. However, the choice of algorithm is not trivial and requires careful consideration of the trade-offs between performance, security, and other requirements.

We discussed the challenges associated with URL shortening, including the need for uniqueness, collision

avoidance, and resilience to attacks. We also described the main goals of URL shortening, such as producing short, easy-to-read, and easy-to-share URLs that preserve the original destination. We then presented three categories of URL shortening algorithms, namely hashing algorithms [12], bijective algorithms, and randomized algorithms, and analyzed several specific examples, such as SHA-2, Base62, and Bloom filter. We compared the performance of these algorithms in terms of speed, memory usage, and collision rate for different lengths of data. By the way, all those algorithms can be used to solve other tasks in areas such as Big Data [13], Artificial Intelligence [14] etc. Our analysis revealed that each algorithm has its own strengths and weaknesses and is suitable for different use cases. For example, hashing algorithms are simple and fast but can suffer from collisions, while bijective algorithms are collision-free but may have limited scalability. Randomized algorithms can provide good trade-offs between speed, uniqueness, and security but require careful tuning.

Overall, our research showed that URL shortening algorithms are a complex and challenging area of study that requires a thorough understanding of the application domain and careful evaluation of the available options.

References

- [1] *Beckett D.; Sezer S., et al.* (2017). HTTP/2 Cannon: Experimental analysis on HTTP/1 and HTTP/2 request flood DDoS attacks, IEEE Xplore. doi: 10.1109/ICME.2007.4284716
- [2] *Yuan, H.; Wun, B.; Crowley, P.; Beckett, et al.* (2010). Software-based implementations of updateable data structures for high-speed URL matching. IEEE Xplore.
- [3] *Wen, S.; Dang, W.; Beckett, D.; Sezer, S.; et al.* (2018). Research on Base64 Encoding Algorithm and PHP Implementation. IEEE Xplore. doi: 10.1109/GEOINFORMAT-ICS.2018.8557068.
- [4] *Arias Aristizábal, L. F., & Duque Méndez, N. D.* (2012). SEO (Search Engine Optimization) schema application for websites with an emphasis on optimizing pages developed in flash. doi: 10.1109/ColombianCC.2012.6398011.
- [5] *Soon L.-K., & Lee S. H.* (2008). Identifying Equivalent URLs Using URL Signatures, IEEE Xplore. doi: 10.1109/SITIS.2008.21.
- [6] *Chen Y., Zhou Y., Dong Q., & Li Q.* (2020). A Malicious URL Detection Method Based on CNN. IEEE Xplore. doi: 10.1109/TOCS50858.2020.9339761
- [7] *Yan, K., Chen, J., Cao, B., Zheng, Y., & Hong, T.* (2014). Research on a low conflict flow matching hash algorithm. IEEE Xplore. doi: 10.1049/cp.2013.2017.
- [8] *Pham H. L., Tran T. H., Le V. T. D., Nakashima Y., & Yan K.* (2022). A Coarse Grained Reconfigurable Architecture for SHA-2 Acceleration: An Improved Low Conflict Flow Matching Hash Algorithm. IEEE Xplore. doi: 10.1109/IP-DPSW55747.2022.00117
- [9] *Kanca A. M., SAĞIROĞLU Ş.* (2021). Sharing Cyber Threat Intelligence and Collaboration: An Improved Low Conflict Flow Matching Hash Algorithm. IEEE Xplore. doi: 10.1049/cp.2021.965432
- [10] *Laatansa, R., Saputra, R., Noranita, B., Yan, K., Chen, J., Cao, B., Zheng, Y., & Hong, T.* (2020). Analysis of GPGPU-Based Brute-Force and Dictionary Attack on SHA-1 Password Hash: An Improved Low Conflict Flow Matching Hash Algorithm. IEEE Xplore. doi: 10.1049/cp.2020.8982390
- [11] *Salamatian S., Huleihel W., Beirami A., Cohen, A., M dard M.* (2020). Centralized vs Decentralized Targeted Brute-Force Attacks: Guessing With Side-Information using an improved low conflict flow matching hash algorithm. IEEE Xplore. doi: 10.1109/TIFS.2020.2998949.
- [12] *Studiawan, H., Pratomo, B. A., Anggoro, R., Yan, K., Chen, J., Cao, B., Zheng, Y., & Hong, T.* (2017). Clustering of SSH brute-force attack logs using k-clique percolation and an improved low conflict flow matching hash algorithm. IEEE Xplore. doi: 10.1109/ICTS.2016.7910269.
- [13] *K. Smelyakov, A. Chupryna, O. Bohomolov and I. Ruban,* "The Neural Network Technologies Effectiveness for Face Detection," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 201-205, doi: 10.1109/DSMP47368.2020.9204049.
- [14] *K. Smelyakov, A. Chupryna, O. Bohomolov and N. Hunko,* "The Neural Network Models Effectiveness for Face Detection and Face Recognition," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476.

The article was delivered to editorial staff on the 20.12.2022

ПРАВИЛА оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ»

Науково-технічний журнал «Біоніка інтелекту» приймає до друку написані спеціально для нього оригінальні рукописи, які раніше ніде не друкувались. Структура рукопису повинна бути такою: індекс УДК, відомості про авторів, заголовок, анотації (на трьох мовах), ключові слова, вступ, основний текст статті, висновки, список використаної літератури, резюме.

Відповідно до Постанови ВАК України від 15.01.2003 №7-05/1 (Бюлетень ВАК, №1, 2003, с. 2), стаття повинна мати такі необхідні елементи: постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями; аналіз останніх досліджень і публікацій і виділення не вирішених раніше частин загальної проблеми в даній області; формулювання цілей та завдань дослідження; виклад основного матеріалу досліджень з повним обґрунтуванням отриманих наукових результатів; висновки з даного дослідження та перспективи подальших досліджень у даному напрямку.

Статті мають бути виконані в редакторі Microsoft Word. Формат сторінки – А4 (210×297 мм), поля: верхнє – 25 мм, нижнє – 20 мм, лівє, правє – 17 мм. Кількість колонок – 2, з інтервалом між ними 5 мм, основний шрифт Times New Roman, кегль основного тексту – 10 пунктів, міжрядковий інтервал – множник (1,1), абзацний відступ – 6 мм. Обсяг рукопису – від 6 до 12 сторінок (мови: українська, англійська, російська та мовою оригінала).

УДК друкується з першого рядка, без відступів, вирівнювання по лівому краю.

ПІБ автора (-ів), назва статті, назва та адреса учбового закладу необхідно надати повністю російською, українською та англійською мовами.

Назва статті друкується прописними літерами; шрифт прямий, напівжирний, кегль 12.

Назви розділів нумерують арабськими цифрами, виділяють жирним шрифтом. Відступи для назви статті, ініціалів та прізвищ авторів, відомостей про авторів, назв розділів, вступу та висновків, списку літератури: зверху – 6 пт, знизу – 3 пт.

Анотації (мовою статті, абзац 6–12 рядків, кегль 9) розміщують на початку статті, в ній має бути розміщена інформація про очікувані результати описаних досліджень (на трьох мовах).

Ключові слова (4–10 слів з тексту статті, які з точки зору інформаційного пошуку несуть змістовне навантаження) наводять мовою рукопису, через кому в називному відмінку, кегль 9.

Рисунки та таблиці (чорно-білі, контрастні) розміщуються у тексті після першого посилання у вигляді окремих об'єктів і нумерують арабськими цифрами наскрізною нумерацією за наявності більше ніж одного об'єкта. Невеликі схеми, що складаються з 3–4 елементів виконують, використовуючи вставку об'єкта Рисунок Microsoft Word. Більш складні виконують у графічних редакторах у вигляді чорно-білих графічних файлів форматів .tif, .jpg, .wmf, .cdr із

розділенням 300 dpi. Рисунки мають міститися у текстовому файлі й обов'язково подаватися окремими файлами з відповідними назвами (наприклад, рис1.jpg).

Усі елементи рисунка, включаючи написи, повинні бути згруповані. Усі написи в рисунках і таблицях мають бути виконані шрифтом Times New Roman, кегль у рисунках – 10, у таблицях – 9.

Рисунок повинен мати центрований підпис (поза рисунком), шрифт 9, відступи зверху і знизу по 6 пт. Ширина рисунка має відповідати ширині колонки (або ширині сторінки).

Формули, символи, змінні повинні бути набрані в редакторі формул **MathType**. Формули розміщують посередині рядка й нумерують за наявності посилань на них у рукописі. Шрифт – Times New Roman. Висота змінної – 10 пунктів, великих і малих індексів – 8 пт, основний математичний символ – 12 (10) пт. Змінні, позначені латинськими літерами, набирають курсивом, грецькі літери, скорочення російських слів і цифри – прямим написанням. Змінні, які є в тексті, також набирають у редакторі формул.

Список літератури вміщує опубліковані джерела, на які є посилання в тексті, укладені у квадратні дужки, друкують без абзацного відступу, кегль 9 пт, відступ зверху – 6 пт.

Після списку літератури з відступом зверху 6 пт зазначають *дату подання статті до редколегії*. Число та місяць задають двозначними числами через крапку. Розмір шрифту – 9 пт, курсив, вирівнювання по правому краю.

Резюме (Times New Roman, кегль – 10 пунктів,) подають англійською мовою: обсяг резюме до 2000 знаків (бажаний переклад). *Структура резюме*: **Background, Materials and methods, Results, Conclusion**.

Разом із рукописом (на аркушах білого паперу формату А4 щільністю 80-90 г/м², надрукований на лазерному принтері) необхідно подати такі документи:

1. Заяву, яку повинні підписати всі автори.
2. Акт експертизи про можливість опублікування матеріалів у відкритому друці (якщо потрібно).
3. Рецензію, підписану доктором чи кандидатом наук.
4. Відомості про авторів.
5. Електронний варіант рукопису, резюме та відомостей про авторів.
6. Зробити оплату публікації.

Необхідно також зазначити один з наступних тематичних розділів, якому відповідає рукопис:

1. Теоретичні основи інформатики та кібернетики. Теорія інтелекту.
2. Математичне моделювання. Системний аналіз. Прийняття рішень.
3. Інтелектуальна обробка інформації. Розпізнавання образів.
4. Інформаційні технології та програмно-технічні комплекси.
5. Структурна, прикладна та математична лінгвістика.
6. Дискусійні повідомлення.

ЗМІСТ

ШТУЧНИЙ ІНТЕЛЕКТ. ОБ'ЄКТНЕ МОДЕЛЮВАННЯ

<i>Shyrovkov V. A.</i> The “Corona” crisis and the “Dark matter” of dollar.....	3
<i>Українець О.Є., Удовенко С.Г., Чала Л.Е., Шергін В.Л.</i> Нейромережева технологія розпізнавання зображень з використанням клітинних автоматів	19
<i>Корабльов М.М., Фомічов О.О., Васюк Д.В.</i> Пошук працездатних гроху-адресів з використанням нейромережевого підходу.....	29
<i>Ляшок Я.О., Подкопаєв С.В., Повзун О.І., Вірич С.О., Калиниченко В.В.</i> Математичне моделювання технологічної системи за допомогою симплекспланування експерименту	35
<i>Четвериков Г.Г., Сикал І.М., Табакова І.С.</i> Деякі аспекти моделювання природномовних процесів та їх застосування при викладанні в вищих навчальних закладах.....	41

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

<i>Revenchuk I., Ostashko Ye.</i> Minimizing the costs of building complex cross-platform applications with Flutter and Firebase	49
<i>Терещенко Г. Ю., Хіміч Є. Р.</i> Порівняльний аналіз традиційних та гібридних моделей сховищ зображень у сфері великих даних.....	58
<i>Revenchuk I., Steshko V.</i> Architectural solutions and optimization methods to improve the performance of Node.js and Vue.js applications	64
<i>Груздо І. В., Кириченко І. В., Тарасов Р. О.</i> Аналіз залежності вибору освітніх альтернативних дисциплін від тенденцій ІТ галузі.....	70
<i>Revenchuk I., Valuiskyi V., Temnokhud M.</i> Wireless methods for the local positioning without access to the Internet and GPS service.....	76
<i>Чалий С.Ф., Лецинський В.О.</i> Можливісно-каузальне представлення пояснень в інтелектуальній інформаційній системі	82
<i>Kupriianov Ye.</i> Theory of L-systems as a framework for creating inflectional e-dictionaries	89
<i>Kyrychenko I., Shamrai Ye.</i> Comparative Analysis of URL Shortening Algorithms: Performance Considerations in the Application Domain of Web Search and Information Retrieval Technology.....	98

ПРАВИЛА

оформлення рукописів для авторів науково-технічного журналу «БІОНІКА ІНТЕЛЕКТУ»	106
---	-----

Наукове видання

БІОНІКА ІНТЕЛЕКТУ
інформація, мова, інтелект

Науково-технічний журнал

№ 1 (98)

2022

Головний редактор — *Г. Г. Четвериков*
Відповідальний редактор — *І. В. Кириченко*

Комп'ютерна верстка — *О. Б. Ісаєва*

Рекомендовано Вченою Радою
Харківського національного університету радіоелектроніки
(протокол № 6 от 22.12.2022)

Адреса редакції:

Україна, 61166, Харків-166, просп. Науки, 14,
Харківський національний університет радіоелектроніки, к. 127
тел. 702-14-77, факс 702-10-13,
e-mail: bionics@nure.ua

Підписано до друку 23.12.2022. Формат 60 × 84 ¹/₈. Друк ризографічний.
Папір офсетний. Гарнітура Newton. Умов. друк. арк. 12,5. Обл.-вид. арк. 12,2.
Тираж 100 прим.

Віддруковано в редакційно-видавничому відділі ХНУРЕ
61166, Харків, просп. Науки, 14.