СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

№ 1(3)/2024 НАУКОВИЙ ЖУРНАЛ ISSN 2788-6603

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

УДК 004
DOI: https://doi.org/10.17721/AIT.2024.1

Подано теоретичні та практичні результати досліджень із програмної інженерії, прикладних інформаційних систем і технологій, мережних та інтернет-технологій, інженерії знань штучного й обчислювального інтелекту, математичних основ інформаційних технологій, інформаційної аналітики й аналітики даних, машинного навчання та розпізнавання образів, цифрових технологій управління ІТ-проєктами, розумних систем та інтернету речей.

Для науковців, фахівців, студентів.

Адреса редколегії

факультет інформаційних технологій
вул. Богдана Гаврилишина, 24, м. Київ, 04116
☎ (38044) 481 45 26
e-mail: ait.knu.fit@gmail.com

Затверджено

вченою радою факультету інформаційних технологій
11.11.24 (протокол № 4)

Зареєстровано

Національною радою України з питань телебачення та радіомовлення
Рішення № 357 від 15.02.24
Ідентифікатор медіа R30-02758

Засновник і видавець

Київський національний університет імені Тараса Шевченка
Видавничо-поліграфічний центр "Київський університет"
Свідоцтво про внесення до Державного реєстру
ДК № 1103 від 31.10.02

Адреса видавця

ВПЦ "Київський університет"
б-р Тараса Шевченка, 14, м. Київ, 01601
☎ (38044) 239 32 22, 239 31 58, 239 31 28
e-mail: vpc@knu.ua

ADVANCED INFORMATION TECHNOLOGY

№ 1(3)/2024 SCIENTIFIC JOURNAL ISSN 2788-6603

TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV

UDC 004
DOI: https://doi.org/10.17721/AIT.2024.1

Theoretical and practical results of research in software engineering, applied information systems and technologies, network and Internet technologies, knowledge engineering, artificial and computational intelligence, mathematical foundations of information technology, information analytics and data analytics, machine learning and pattern recognition, digital management technologies of IT projects, smart systems and the Internet of Things are presented.

For scientists, specialists, students.

Address	**Faculty of Information Technology** **24, Bohdan Hawrylyshyn str., Kyiv, 04116** **☎ (38044) 481 45 26** **e-mail: ait.knu.fit@gmail.com**
Approved by	**the Academic Council of the Faculty of Information Technology** **11.11.24 (protocol № 4)**
Registered by	**the National Council of Television and Radio Broadcasting of Ukraine** **Decision № 357 of 15.02.24** **Media identifier R30-02758**
Founder and publisher	**Taras Shevchenko National University of Kyiv** **Publishing and Polygraphic Center "Kyiv University"** **Certificate of entry into the State Register** **ДК № 1103 dated 31.10.02**
Address	**PPC "Kyiv University"** **14, Taras Shevchenka blvd., Kyiv, 01601** **☎ (38044) 239 32 22, 239 31 58, 239 31 28** **e-mail: vpc@knu.ua**

ЗМІСТ

Прикладні інформаційні системи та технології

Мережні й інтернет-технології

CONTENTS

Applied information systems and technology

Network and internet technologies

ПРИКЛАДНІ ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

УДК 004.89
DOI: https://doi.org/10.17721/AIT.2024.1.01

Maiia BOCHAROVA, PhD Student
ORCID ID: 0009-0004-3875-5019
e-mail: bocharova.maiia@gmail.com
Odesa I. Mechnikov National University

Eugene MALAKHOV, DSc (Engin.), Prof.
ORCID ID: 0000-0002-9314-6062
e-mail: eugene.malakhov@onu.edu.ua
Odesa I. Mechnikov National University

GENERAL-PURPOSE TEXT EMBEDDINGS LEARNING FOR UKRAINIAN LANGUAGE

Background. ***Learning high-quality text embeddings typically requires large corpuses of labeled data, which can be challenging to obtain for many languages and domains. This study proposes a novel adaptation of cross-lingual knowledge transfer that employs a cosine similarity-based loss calculation to enhance the alignment of learned representations.***

Methods. ***The impact of teacher model selection on the quality of learned text representations is investigated. Specifically, the correlation between cosine similarity scores among vectors of randomly selected sentences and the transferability of representations into another language is explored. Additionally, recognizing the need for effective evaluation methodologies and the limited availability of Ukrainian resources within existing benchmarks, a comprehensive general-purpose benchmark for assessing Ukrainian text representation learning is curated.***

Results. ***A cosine-similarity based loss calculation leads to 14.2% improvement in absolute Normalized Mutual Information (NMI) score compared to using mean squared error loss when distilling knowledge from the English language teacher model into Ukrainian student model. The findings demonstrate the strong correlation between the distributions of cosine similarities of the teacher model's representations of random sentences with the quality of learnt text embeddings. Pearson's correlation between "90th percentile of cosine similarity scores distribution" and "Average NMI score" is -0.96, which is a strong negative correlation.***

Conclusions. ***This research advances information theory in cross-lingual knowledge distillation, illustrating that cosine similarity-based loss functions are superior in performance. It underscores the importance of selecting the teacher model with wide distributions of cosine similarity scores. Furthermore, a pioneering broad-scale benchmark, covering five distinct domains for Ukrainian text representation learning is introduced. The source code, pretrained model, and the newly created Ukrainian text embeddings benchmark are publicly available at https://github.com/maiiabocharova/UkrTEB.***

Keywords: ***Natural Language Processing, text embeddings, Deep Learning, Data Mining, multilingual language models, knowledge transfer, domain adaptation.***

Background

There is an ever-growing interest in text representation learning, since it offers an attractive method to not only explore large text collections and find groups of semantically similar documents (Filatov, & Kovalanko, 2020), recommend relevant documents to users, but also allows to augment the input to Large Language Models (LLMs) (Binder, & Mezhuyev, 2024), such as Retrieval Augmented Generation frameworks (RAGs), enhancing their ability to generate relevant responses (Guu et al, 2020).

Models for creating text representations in vector space like SBERT (Reimers, & Gurevych, 2019), E5 (Wang et al., 2022), and GTE (Li et al., 2023) have demonstrated the strong capabilities of specialized transformer-based models to learn text embeddings of excellence in the English language, exploiting the vast availability of high-quality datasets specifically curated for semantic similarity learning, among which NLI (Bowman et al., 2015) and SNLI (Williams et al., 2018).

Recently, a lot of researchers have been focusing on unsupervised text representation learning, using contrastive learning and diverse augmentations to construct positive text pairs. However, while not requiring labeled data, those approaches tend to perform worse than models trained with supervision (Wang, Reimers, & Gurevych, 2021). This means that learning high-quality text embedding usually requires large corpuses of labeled data, which for low-resource languages are not available. Taking into account availability of a large number of strong models capable of producing high-quality text embeddings for English language, along with the translated text-pairs, one possible solution is to distill the knowledge of the teacher English model into a student model which will work for a low-resource language. Specifically, cross-lingual alignment of representations allows to transfer knowledge learnt by the model in one (source) language to another (target) language without the need for specific data annotation in the target language.

Using neural language models to produce embeddings for getting text representation is becoming more and more popular, replacing classical "bag-of-words" approaches. The vast availability of frameworks for training and inferencing neural embeddings models only contributes to the trend. Among such frameworks SentenceTransformers library offers a vast collection of pretrained transformer-based models and techniques to either train from scratch or fine-tune existing text embedding models.

Benchmarks are especially useful in comparing the benefits of training the models and assessing how beneficial it is to use one or another data source or approach for training.

The Massive Text Embeddings Benchmark (MTEB) (Muennighoff et al., 2022) offers a wide range of unified tasks (e.g. summarization, reranking, clustering, pair classification, retrieval etc) and domains (news, scientific publications, reviews, comments in social media etc) for assessing embeddings in different languages. However, MTEB mainly focuses on English datasets (97 out of 185 available datasets in it contain English texts) and Chinese (43). Only 19 languages occur more than 4 times. Ukrainian, among other languages is very underrepresented, being part of only 2 datasets which are both part of the bitext mining task – namely test set of the "Tatoeba" (Tiedemann, 2020) dataset, which is a collection of crowdsourced by volunteers sentences and their corresponding translations; and Flores (Goyal, 2022), which is a benchmark dataset compiled by Meta for machine translation between English and low-resource languages.

Considering the scarcity and specialized nature of datasets available for Ukrainian language, which are focused on translation tasks, there is a necessity to establish a benchmark that could help to accurately evaluate the performance of text embeddings in Ukrainian language across various domains.

ISSN 2788-6603

Aim formulation. The improvement of the information theory for cross-lingual knowledge distillation in text representation learning for low-resource languages on the example of Ukrainian language, leveraging English – target language translated pairs and state-of-the-art English teacher model.

Introduction of a new benchmark for assessing performance of text representation learning models for Ukrainian language, containing diverse domains.

Related Works. *Benchmarks.* The practice of web crawling and data mining to collect categorized data is known to play an important role in creating the benchmarks. As such, among many others, governmental job boards were scraped to create title normalization benchmarks (Decorte et al., 2021), news documents were extracted (Lang, 1995) to make clustering and classification dataset; and categorized question titles from StackExchange were extracted (Geigle et al., 2021).

MTEB (Muennighoff et al., 2022], having unified a substantial number of different datasets into a one benchmark, has established itself as the gold-standard for evaluating English models. Furthermore, efforts for extending MTEB to some other languages have been undertaken. As such, among others, C-Pack (Xiao et al., 2023) – a package of resources for benchmarking Language embedding for Chinese language, German Text Embeddings clustering benchmark (Wehrli, , Arnrich, & Irrgang, 2023) for assessing capabilities of models for grouping German texts and Spanish Evaluation benchmark (Araujo et al., 2022) for measuring performance of Spanish language embedding models were introduced.

The overall aim of such benchmarks is to allow a fair and reproducible assessment between models.

Text embeddings models. With the introduction of BERT (Kenton, & Toutanova, 2019), the new era of neural textual representations has begun, setting new state-of-the-arts on different text processing benchmarks. However, raw BERT representations are not suitable for providing efficient text-level embeddings, due to its training objectives (Reimers, Gurevych, 2019). To overcome this limitation, a number of approaches were proposed for adapting BERT embeddings to make them representative on sentence or paragraph-level. As such Universal Sentence Encoder (Cer et al., 2018) and Sentence BERT (Reimers Reimers, Gurevych, 2019) models were introduced, trained with supervision leveraging large human-labeled datasets (Bowman et al., 2015; Williams et al., 2018).

The growing interest towards multilingual text embedding learning and bitext mining for training Neural Machine Translation (NMT) systems has led to a large number of proposed models and architectures (Artetxe, & Schwenk, 2019; Heffernan, Çelebi, & Schwenk,2022) for language-agnostic representation learning. Most of the works focus on parallel text pairs, authors of EASE (Nishikawa et al., 2022) explore the entity linking in wikipedia to construct pairs of related texts. Authors of (Feng et al., 2020; Artetxe et al., 2019) prove the beneficial effects of such learning for the languages with very limited resources, originating from the fact that during multilingual training the model might have seen similar languages.

However, as is widely known (Xu et al., 2023), training a multilingual model presents a challenge of language interference, and, when having sufficient monolingual data, specialized monolingual models tend to outperform their multi-lingual same-sized counterparts.

And, to the best of our knowledge, no specialized models for creating sentence embeddings for Ukrainian language have been released. This can be attributed to several implications, such as lack of well-prepared datasets for training and benchmarks to evaluate the capabilities of the models.

To address the challenge described above we create a specialized benchmark for Ukrainian text embeddings learning – UkrTEB, as well as study the available resources for learning Ukrainian text representations and present a novel approach for training general-purpose text embeddings model for Ukrainian language (https://github.com/maiiabocharova/UkrMTEB). Our method leverages recent advancements in transfer learning techniques to overcome the challenges posed by the limited availability of labeled data and computational resources.

Methods

Dataset collection. Following the MTEB's design for Ukrainian texts, aim is to ensure high diversity of collected data, allowing to simulate a broad range of real-world applications. To address this problem of diversity, different domains offering both formal and informal texts should be considered. Furthermore, the usage of the data of permissive nature should be ensured.

In the context of the average sample length diversity across various datasets, MTEB typically incorporates two distinct versions of a dataset originating from a single data source. These versions are tailored to accommodate differences in text length, with one version focusing on sentence-to-sentence comparisons for shorter texts and another version geared towards paragraph-to-paragraph comparisons intended for longer texts. Specifically, taking as an example data gathered from StackExchange, two datasets are compiled. The first dataset presents the headlines of questions, whereas the second dataset includes both the headline and its accompanying description to the models for analysis.

1. Ukrainian government legislation board

The first data source chosen for analysis is the Ukrainian government legislation board. This entails the extraction of draft law titles along with their associated categories. Nineteen of the most frequently occurring categories were selected for the benchmark, including those related to committees addressing "правоохоронна діяльність" (law enforcement activities), "фінанси, податкова та митна політика" (finance, tax, and customs policy), among others. This selection process yielded a corpus comprising 9,678 samples, with an average length of 26.1 words per sample.

2. Book titles with their descriptions

The second data source encompasses book titles, accompanied by their blurbs and corresponding categories, amounting to a total of 12,590 samples across 53 distinct categories. This dataset is distinct from the first, focusing on literary works spanning a wide array of categories. These entries encompass a broad spectrum of themes and subjects, ranging from technical domains such as "Комп'ютерна література" (computer literature) to more nuanced explorations of human psychology and relationships in "Психологія і взаємини" (psychology and relationships), as well as imaginative and fantastical narratives designed for children. Each sample within this dataset comprises an average of 126.8 words, making it possible to test the models' capabilities for handling longer paragraphs.

3. Petitions to the ukrainian government

The third source of data is a website dedicated to hosting petitions. This platform serves as a digital space where individuals can create and sign petitions on various social, political, and environmental issues. Unlike the preceding datasets focused on legislative documents and literary works, this source captures public sentiment and activism, providing insights into societal

concerns and advocacy efforts. This dataset encompasses the collection of 3,387 samples distributed across 18 distinct categories. Through the aggregation of petitions, this dataset offers a diverse array of topics, reflecting the multitude of interests and causes championed by online communities. Analyzing this data enables researchers to examine trends in public opinion, track emerging issues, and assess the efficacy of grassroots activism in shaping public discourse and policy agendas.

4. UkrQAForum

The fourth data source selected for analysis is a Ukrainian community question answering forum, chosen to capture authentic conversational language usage and cover a diverse array of topics. This dataset aims to provide insights into real-world communication patterns, reflecting the colloquial language and informal discourse commonly found in online community forums. Since users post questions not only in Ukrainian, but also in other slavic languages, the FastText language model is used to filter out non-Ukrainian language data. The dataset contains 86 distinct categories, among which "Стиль, Мода, Бренди > Бренди" (Style, fashion, brands > Brands), "Сім'я, Дім, Діти > Домашні тварини" (Family, home, children > House Pets) etc.

Like real-world texts there are grammatical mistakes, typos and surzhyk (mixture of Ukrainian and other languages, spoken in the region)

5. UkrNews

The dataset collected for the benchmark consists of the Ukrainian news articles sourced from the biggest Ukrainian news outlet. This addition to the data sources aims to provide a comprehensive view of language usage across journalism and media domains. News articles offer a distinct perspective on language usage, characterized by formal structures, journalistic conventions, and specialized terminology.

This dataset contains 23 categories, among which "війна" (war), "бокс" (boxing).

Some examples of the data samples are shown in Table 1.

Table 1

Example samples from the collected datasets

Data source	Sample	Category of the sample
Ukrainian government legislation board	Проєкт Закону про внесення змін до Закону України "Про адвокатуру та адвокатську діяльність" щодо вдосконалення окремих питань організації адвокатської діяльності	Правова політика
Book titles with their descriptions	"Легкі й швидкі рецепти неперевершених десертів для святкового дня і просто гарного настрою! Ніжний торт, ароматний лимонний чізкейк, рум'яні кексики, пиріжки, еклери… Понад 50 рецептів з оригінальними фотографіями надихнуть вас на солодкі експерименти на кухні!"	Кулінарія. Їжа та напої
Petitions to the ukrainian government	"Про порядок повірок квартирних/будинкових лічильників води"	Комунальне господарство
QA Forum	"Київ, 2-кімнатна в новобудові – по чому?"	Бізнес, Фінанси. Нерухомість
News	"День фізичної культури і спорту в Україні: листівки та побажання"	Свята

The aggregated statistics of the benchmark datasets are shown in Table 2.

Table 2

Statistics of the collected datasets

Dataset Name	Size per split	Classes per split	Number of categories	Number of samples	Average sample length (chars)	Average word lengths
UkrLawDrafts	9,687	19	19	9,687	186.8	26.1
UrkBookBlurs	1250 to 1693	11 to 15	48	16,678	748.4	123.7
UkrPetitions (s2s)	3,474	18	18	3,474	75.75	10.8
UkrPetitions (p2p)	3,474	18	18	3,474	1332	197
UkrQAForum (s2s)	4380 to 8475	11 to 15	86	155,788	42,3	7.9
UkrQAForum (p2p)	4380 to 8375	11 to 15	86	155,781	311	56.9
UkrNews (s2s)	8,159 to 8,457	15	23	74,746	72.3	12.3
UkrNews (p2p)	8,159 to 8,457	15	23	74,746	152.4	25.4

Where s2s means that only headline sentences are taken as samples, and in the p2p scenario respectively the concatenated headline and description paragraphs are used.

Training ukrainian sentence embeddings model. Vast amounts of parallel text chunks which offer datasets like OpenSubtitles (Lison, & Tiedemann, 2021) or wikimatrix (Schwenk et al., 2019) have been collected and made publicly available.

ISSN 2788-6603

A number of methodologies for learning multilingual sentence representation models were proposed, from which two main families of approaches for training embedding models leveraging translated text pairs can be identified.

The first one consists in training a dual encoder model using translation ranking loss with inbatch negative samples (Feng et al., 2020). This training task consists of aligning the embeddings of the source and target languages in a shared space. Models trained with this approach learn to project the embeddings of the same sentence in different languages close to each other, while putting the representation of other sentences which are not direct translation of each other further apart. However, as has been shown by multiple studies (Wang et al., 2022) that, while the strategy of using inbatch negative samples is quite efficient, it requires substantial batch sizes to reach optimal performance, and bigger batch sizes tend to be computationally intensive. The second drawback of such a training strategy lies in the fact that models trained in such a way tend to struggle with estimating the similarity of sentences which are not direct translations of each other (Reimers, & Gurevych, 2020).

The second approach involves utilizing a teacher model proficient in representing sentences in a language with abundant resources. The student model is then trained to emulate the representations generated by the teacher model for the target language text and produce embeddings closely aligned with those of the teacher model in the vector space (Reimers, Gurevych, 2020]. This approach mitigates the influence of batch size on the training process, ensuring that the quality of embeddings learned by the student model depends primarily on the abilities of the teacher's models and the size of the dataset, as well as its relevance to the texts which the student model will encounter while solving downstream tasks.

However, models trained for a specific language tend to outperform multilingual models of the same size.

In LASER3 (Heffernan et al., Tiedemann 2022), for example, researchers propose to use a language family specific model to both benefit from increased numbers of samples and transfer additional knowledge from other similar languages.

Bitext data aligned with rich-resource language (English in our case) is required to train the student Ukrainian text representation model. Training data is collected from the Opus website (Tiedemann, 2012] and consists of a combination of several corpuses. Since the inconsistent quality of the data and language contamination, language deduplication and heuristics-based cleaning is applied.

Specifically texts are cleaned using the following steps:

1. Deduplication.
2. Language identification and filtering of samples which have non-Ukrainian target texts.
3. Filtering out the samples that differ in length by more than two times.

The statistics of the combined dataset are provided in Table 3.

Table 3

Statistics of the bitext datasets used for training the model

Data Source	Number of parallel sentences	Number of unique sentences after filtering
OpenSubtitles2018	877,780	419,004
Wikimedia	757,910	617,809
SciPar_Ukraine	306,813	306,813
QED	215,530	198,100
TED2020	208,141	198,876
Tatoeba	175,502	168,480
ELRC-5179-acts_Ukrainian	129,942	126,361
ELRC-5180-Official_Parliament_Ukraine_Ukrainian_laws_EN	116,260	115,494
ELRC-5181-Official_Parliament_Ukraine_abstracts_UK_laws	61,012	60,885
ELRC-5174-French_Polish_Ukraine	36,228	35,597
Total	2,885,118	2,264,340 Globally unique: 2,202,117

After careful consideration we opt out from using the automatically mined sentence pairs like WikiMatrix dataset, because upon manual inspection it is discovered that such datasets are of very low quality for English-Ukrainian language pairs (e.g. "And they ask you what they should spend" is a pair of "Потім запитали: чи не ти пророк?", "How are you feeling?" with "Який спосіб запропонували б ви?" etc), which underscores the lack of well-performing models for ukrainian language.

After final deduplication the obtained dataset contains 2,202k bitexts.

In (Reimers, & Gurevych, 2020) authors propose to use Mean Squared Error (MSE) as a loss function. MSE is a commonly used loss function in regression tasks and measures the average squared difference between the predicted and actual values. In the context of text representation learning, MSE compares the embeddings produced by the student model with those generated by the teacher model, aiming to minimize the discrepancy between the two sets of embeddings. However, training with MSE loss can lead to unstable training and unexpected results, especially when the embeddings of the teacher model are normalized and MSE values are small. Moreover, cosine similarity is used for the final measurement of similarities when handling downstream tasks.

Cosine similarity measures the cosine of the angle between two vectors and is particularly well-suited for comparing the similarity between vectors regardless of their magnitudes. In the context of text embeddings, cosine similarity quantifies the degree of similarity between two text representations based on the direction of their vectors in the embedding space, rather than their absolute distances and as such is more robust in capturing semantic similarity between text representations.

Taking the above into account, we adapt the M-SBERT (Reimers et al., 2020) training approach to use cosine similarity loss for training.

More formally, loss is defined as follows:

$$\text{Loss} = (1 - \cos(emb_1, emb_2))^2 , \quad (1)$$

where emb_1 – is embedding of the English sentence, produced by teacher model; emb_2 – is embedding of the Ukrainian sentence, produced by student model; cos – is a cosine similarity between two vectors.

To save computational resources we initialize the weights of the ukrainian language model from the RoBERTa-base model trained on ukrainian text data (ukr-roberta-base, 2024).

AdamW optimizer with a linear warmup for 10% of steps is used. Learning rate is set to 2e-5.

Results

For evaluation three publicly available models supporting Ukrainian language are taken. As evaluation metric, the Normalized Mutual Information (NMI) score, which is a harmonic mean of homogeneity and completeness, is used. This score can be in range 0 to 1, where 1 means a perfectly complete labeling of samples. This metric is not dependent on the absolute values of the labels and permuting the class or cluster label values does not affect the score.

The evaluation results are presented in Table 4.

Table 4

Evaluation results of NMI score

Model	UkrLawDrafts	UrkBookBlurs	UkrPetitions		UkrQAForum		UkrNews		Average
LaBSE	0,175	0,513	0,138	0,249	0,188	0,396	0,559	0,585	0,350
distiluse-base-multilingual-cased	0,138	0,410	0,162	0,222	0,144	0,326	0,416	0,485	0,289
E5	0,201	0,481	0,205	0,274	0,300	0,459	0,627	0,644	0,40
Ukr-distil_mse	0,145	0,329	0,122	0,121	0,133	0,207	0,517	0,602	0,272
Ukr-distil_cos	0,234	0,542	0,246	0,326	0,326	0,453	0,568	0,621	0,414

As can be seen from the results, there is a clear advantage of using cosine similarity loss over 0,272 average score, which accounts mean squared error loss (0,414 over to 14.2% absolute NMI score score improvement).

Dependence of quality of learn text embeddings on teacher model. Experiments using different teacher models were conducted. Below in fig. 1 the distributions of the cosine similarities calculated between all possible pairs of vectors from 5,000 random English sentences from the training set are visualized. As can be seen, E5 model shows a very narrow distribution.

The plot (Fig. 1) shows cumulative distribution of cosine similarity scores, where the x-axis represents the cosine similarity threshold, and the y-axis shows the percentage of records that have a similarity score less than that threshold

Correlation between cosine similarity scores distribution of the teacher model and performance on the downstream tasks of the student model distilled from it is shown in table 5.

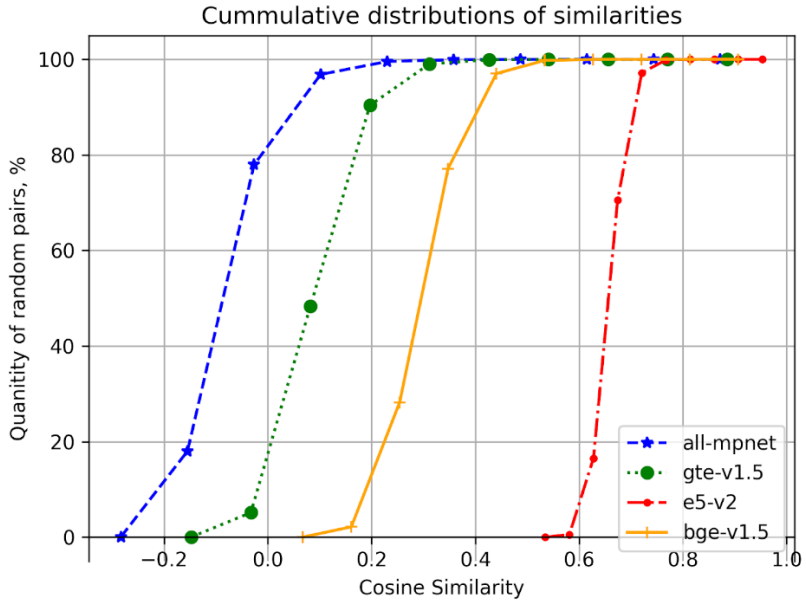


Fig. 1. Cumulative distributions of similarity scores between random sentence pairs

ISSN 2788-6603

Table 5

Correlation between cosine similarity distribution scores and knowledge transferability

Teacher model	UkrLawDrafts	UrkBookBlurs	UkrPetitions		UkrQAForum		UkrNews		Average score	90th percentile of cosine similarity
e5 base	0,062	0,245	0,054	0,045	0,065	0,127	0,308	0,414	0,165	0,75
bge base	0,113	0,342	0,124	0,112	0,133	0,215	0,516	0,601	0,269	0,48
gte base	0,128	0,364	0,105	0,199	0,132	0,263	0,448	0,548	0,273	0,31
all-mpnet	0,234	0,542	0,246	0,326	0,326	0,453	0,568	0,621	0,414	0,16

As can be seen from Table 5, there exists a strong correlation between the distribution of cosine similarities between vectors of unrelated sentences and quality of the sentence embeddings learnt by student model. Pearson's correlation between "90th percentile of cosine similarity" and "Average score" is – 0.96, which is interpreted as a strong negative correlation.

The lower similarity between unrelated vectors leads to more stable training and learning by the student model of more distinct sentence embeddings, which in turn enhances the model's ability to accurately capture the semantic nuances of the text.

This results in improved performance on downstream tasks, as the embeddings are able to better represent the underlying information and differentiate between various content types. Therefore, selecting a teacher model with a lower average cosine similarity score is crucial for effective knowledge transfer and the overall success of the student model.

Discussion and conclusion

In this paper information theory for cross-lingual knowledge distillation obtained further development. It has been shown that using cosine similarity-based loss function leads to significant improvements (14.2% absolute NMI score improvement) compared to using the mean squared loss function when distilling knowledge from the well-trained model.

The influence of the teacher model selection and correlation between the cosine similarity distribution which the teacher model produces was proved to influence the quality of learn embeddings of the student model. It has been established that there is a strong negative correlation between "90th percentile of cosine similarity scores distribution" and "Average NMI score" obtained on clustering benchmark.

A new and first of its kind benchmark for Ukrainian text representation learning has been introduced, covering 5 distinct domains.

The model and Ukrainian text embeddings benchmark are freely available at https://github.com/maiiabocharova/UkrMTEB.

Authors' contribution. Maiia Bocharova – literature overview, development of methods and methodologies of the research, empirical data collection, analysis of results and conclusions. Eugene Malakhov – consultation, ideas and guidance.

References

Abdelali, A., Guzman, F., Sajjad, H., & Vogel, S. (2014). The AMARA Corpus: Building parallel language resources for the educational domain. In N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.).*The Proceedings of the 9th International Conference on Language Resources and Evaluation* (pp. 1856–1862)/ In Lrec.

Araujo, V., Carvallo, A., Kundu, S., Cañete, J., Mendoza, M., Mercer, R. E., & Soto, A. (2022). Evaluation Benchmarks for Spanish Sentence Representations. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, S. Piperidis (Eds.). *In Proceedings of the Thirteenth Language Resources and Evaluation Conference* (pp. 6024–6034), Marseille, France. European Language Resources Association.

Artetxe, M., & Schwenk, H. (2019). Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. In L. Lee, M. Johnson, B. Roark, A. Nenkova (Eds.). *Transactions of the Association for Computational Linguistics*, 7, 597–610. https://doi.org/10.1162/tacl_a_00288

Binder, M., & Mezhuyev, V. (2024). *A framework for creating an IoT system specification with ChatGPT*. Internet of Things, 27, 101218. Institute of Industrial Management, University of Applied Sciences FH JOANNEUM, Austria. https://doi.org/10.1016/j.iot.2024.101218

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In L. Màrquez, C. Callison-Burch, J. Su (Eds.). *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 632–642). https://doi.org/10.18653/v1/D15-1075

Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., & Kurzweil, R. (2018). Universal sentence encoder. In E. Blanco, W. Lu (Eds.). *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 169–174). Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-2029

Decorte, J. J., Van Hautte, J., Demeester, T., & Develder, C. (2021). Jobbert: Understanding job titles through skills. *In International workshop on Fair, Effective And Sustainable Talent management using data science (FEAST)* (pp. 1–9). As part of ECML-PKDD.

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2020). Language-agnostic BERT sentence embedding. In S. Muresan, P. Nakov, A. Villavicencio (Eds.). *In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, 1 (Long Papers)* (pp. 878–891). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.62

Filatov, V., & Kovalenko, A. (2020). Fuzzy systems in data mining tasks. *In Advances in Spatio – Temporal Segmentation of Visual Data* (pp. 243–274). Springer. https://doi.org/10.1007/978-3-030-35480-0_6

Geigle, G., Reimers, N., Rücklé, A., & Gurevych, I. (2021). *TWEAC: transformer with extendable QA agent classifiers*. https://doi.org/10.48550/arXiv.2104.07081

Goyal, N., Gao, C., Chaudhary, V., Chen, P. J., Wenzek, G., Ju, D., & Fan, A. (2022). The flores-101 evaluation benchmark for low-resource and multilingual machine translation. In B. Roark, A. Nenkova (Eds.). *Transactions of the Association for Computational Linguistics,* 10, 522–538. https://doi.org/10.1162/tacl_a_00474

Grabar, N., & Hamon, T. (2017). Creation of a multilingual aligned corpus with Ukrainian as the target language and its exploitation. *In Computational linguistics and intelligent systems (COLINS 2017)*. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M. (2020). Retrieval augmented language model pre-training. *In International conference on machine learning* (pp. 3929–3938). JMLR.org.

Heffernan, K., Çelebi, O., & Schwenk, H. (2022). Bitext mining using distilled sentence representations for low-resource languages. In Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.). *Findings of the Association for Computational Linguistics: EMNLP* (pp. 2101–2112), Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.154

Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, T. Solorio (Eds.). *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 (Long and Short Papers)* (pp. 4171–4186). https://doi.org/ 10.18653/v1/N19-1423

Lang, K. (1995). Newsweeder: Learning to filter netnews. *Proceedings of the 12th International Conference on Machine Learning* (pp. 331–339).

Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., & Zhang, M. (2023). Towards general text embeddings with multi-stage contrastive learning. https://doi.org/10.48550/arXiv.2308.03281

Lison, P., & Tiedemann, J. (2016). *Opensubtitles2016:* Extracting large parallel corpora from movie and tv subtitles. N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.). *Proceedings of the 10th International Conference on Language Resources and Evaluation* (pp. 923–929). In Lrec.

Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2022). MTEB: Massive text embedding benchmark. In A. Vlachos, I. Augenstein (Eds.). *In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 2014–2037). Dubrovnik, Croatia. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.eacl-main.148

Nishikawa, S., Ri, R., Yamada, I., Tsuruoka, Y., & Echizen, I. (2022). EASE: Entity-aware contrastive learning of sentence embedding. In M. Carpuat, M. Marneffe, I. Ruiz (Eds.). *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 3870–3885). Association for Computational Linguistics.

Reimers, N. & Gurevych, I. (2019). SentenceBERT: Sentence embeddings using siamese BERT networks. In K. Inui, J. Jiang, V. Ng, X. Wan (Eds.). *Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Hong Kong: China (pp. 3982–3992). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410.

Reimers, N., & Gurevych, I. (2020). Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. In B. Webber, T. Cohn, Y. He, Y. Liu (Eds.). *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 4512–4525), online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.365

Schwenk, H., Chaudhary, V., Sun, S., Gong, H., & Guzmán, F. (2019). Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 1351–1361). Association for Computational Linguistics.

Schwenk, H., Wenzek, G., Edunov, S., Grave, E., & Joulin, A. (2021). CCMatrix: Mining billions of high-quality parallel sentences on the web. In P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.).*In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 1 (Long Papers)* (pp. 6490–6500), online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.115

Tiedemann, J. (2012). *Parallel data, tools and interfaces in OPUS* (pp. 2214–2218). In Lrec.

Tiedemann, J. (2020*).* The Tatoeba Translation Challenge--Realistic Data Sets for Low Resource and Multilingual MT. In L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, Y. Graham, P. Guzman, Ba. Haddow, M. Huck, A. Yepes, P. Koehn, A. Martins, M. Morishita, C. Monz, M. Nagata, T. Nakazawa, M. Negri (Eds.). *In Proceedings of the Fifth Conference on Machine Translation* (pp. 1174–1182). Association for Computational Linguistics.

"ukr-roberta-base" (11, August, 2024). https://huggingface.co/youscan/ukr-roberta-base

Wang, K., Reimers, N., & Gurevych, I. (2021). TSDAE: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning. M. Moens, X. Huang, L. Specia, S. Yih (Eds.). *In Findings of the Association for Computational Linguistics: EMNLP*. Punta Cana, Dominican Republic (pp. 671–688). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.59

Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., & Wei, F. (2022). *Text embeddings by weakly-supervised contrastive pre-training*. https://doi.org/10.48550/arXiv.2308.03281

Wehrli, S., Arnrich, B., & Irrgang, C. (2023). German Text Embedding Clustering Benchmark. In M. Georges, A. Herygers, A. Friedrich, B. Roth (Eds.). *In Proceedings of the 19th Conference on Natural Language Processing.* Ingolstadt, Germany (pp. 187–201). Association for Computational Linguistics.

Williams, A., Nangia, N., & Bowman, S. R. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In M. Walker, H. Ji, A. Stent (Eds.). *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 (Long Papers)* (pp. 1112–1122). Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1101

Xiao, S., Liu, Z., Zhang, P., & Muennighof, N. (2023). *C-pack: Packaged resources to advance general chinese embedding*. https://doi.org/10.48550/arXiv.2309.07597

Xu, H., Tan, W., Li, S. S., Chen, Y., Van Durme, B., Koehn, P., & Murray, K. (2023). Condensing Multilingual Knowledge with Lightweight Language-Specific Modules. In H. Bouamor, J. Pino, K. Bali (Eds.). *In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 1575–1587). https://doi.org/10.18653/v1/2023.emnlp-main.97

Отримано редакцією журналу / Received: 08.09.24
Прорецензовано / Revised: 14.10.24
Схвалено до друку / Accepted: 23.10.24

Майя БОЧАРОВА, асп.
ORCID ID: 0009-0004-3875-5019
bocharova.maiia@gmail.com
Одеський національний університет імені І. І. Мечникова

Євгеній МАЛАХОВ, д-р техн. наук, проф.
ORCID ID: 0000-0002-9314-6062
eugene.malakhov@onu.edu.ua
Одеський національний університет імені І. І. Мечникова

ТРЕНУВАННЯ ТЕКСТОВИХ ВКЛАДЕНЬ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ ДЛЯ УКРАЇНСЬКОЇ МОВИ

В с т у п . ***Тренування високоякісних текстових вкладень зазвичай вимагає великих корпусів з анотованими даними, які може бути складно отримати для більшості мов і доменів. У цьому дослідженні запропоновано нову адаптацію крос-лінгвістичного перенесення знань, яка використовує обчислення втрат на основі косинусної подібності між перекладами текстів для кращого зіставлення отриманих векторних представлень текстів.***

М е т о д и . ***Досліджено вплив функцій втрат, а також вибору моделі вчителя на якість вивчених текстових репрезентацій. Крім того, досліджено кореляцію між розподілом косинусної подібності між векторами випадково вибраних речень моделі-вчителя та можливістю перенесення репрезентацій на іншу мову. З огляду на потребу в ефективних методологіях оцінювання й обмежену доступність ресурсів для української мови в межах існуючих бенчмарків, розроблено комплексний універсальний бенчмарк для оцінювання представлень тексту для української мови.***

Р е з у л ь т а т и . ***Обчислення втрат на основі косинусної подібності приводить до покращення абсолютного показника нормалізованої взаємної інформації (NMI) на 14,2 % порівняно з використанням середньоквадратичної функції втрат під час перенесення знань із моделі-вчителя англійської мови на українську модель-учня. Отримані результати демонструють сильну кореляцію між розподілом косинусної подібності векторів не пов'язаних між собою речень, які векторизуються моделлю-вчителем, та якістю засвоєних текстових вкладень. Кореляція Пірсона між "90-м процентилем розподілу оцінок косинусної подібності" та "середнім показником NMI" становить –0,96, що є сильним негативним зв'язком.***

В и с н о в к и . ***Це дослідження розвиває теорію інформації в галузі крос-лінгвістичної дистиляції знань, показуючи, що функції втрат на основі косинусної подібності є кращими за своїми характеристиками. Підкреслено важливість вибору моделі-вчителя із широким розподілом коефіцієнта косинусної подібності. Представлено новий широкомасштабний бенчмарк, що охоплює п'ять різних доменів для навчання представлення українського тексту. Код, попередньо навчена модель і новостворений бенчмарк для української мови опубліковано за посиланням https://github.com/maiiabocharova/UkrTEB.***

К л ю ч о в і с л о в а : ***оброблення природної мови, текстові вкладення, глибоке навчання, видобування даних, багатомовні мовні моделі, перенесення знань, адаптація до домену.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

УДК 004.896
DOI: https://doi.org/10.17721/AIT.2024.1.02

Наталія АКСАК, д-р техн. наук, проф.
ORCID ID: 0000-0001-8372-8432
e-mail: nataliia.axak@nure.ua
Харківський національний університет радіоелектроніки, Харків, Україна

Максим КУШНАРЬОВ, канд. техн. наук
ORCID ID: 0000-0002-3772-3195
e-mail: maksym.kushnarov@nure.ua
Харківський національний університет радіоелектроніки, Харків, Україна

Юрій ШЕЛІХОВ, асп.
ORCID ID: 0009-0009-8970-6571
e-mail: yurii.shelikhov@nure.ua
Харківський національний університет радіоелектроніки, Харків, Україна

ІНТЕЛЕКТУАЛЬНЕ КЕРУВАННЯ МІКРОКЛІМАТОМ СІТІ-ФЕРМИ НА ОСНОВІ АЛГОРИТМУ Q-LEARNING

В с т у п . ***У контексті швидкого розвитку сіті-фермерства та зростання інтересу до сталого виробництва харчових продуктів, керування мікрокліматом стає ключовим аспектом для досягнення оптимального вирощування рослин. Оптимальне керування температурою, вологістю та освітленням може допомогти використовувати обмежений простір ефективніше, збільшуючи врожайність на одиницю площі. Системи контролю клімату, які дозволяють створювати оптимальні умови для рослин, дозволяють збільшити виробництво на обмеженій площі. Метою дослідження є прийняття обґрунтованих рішень у системі контролю клімату, заснованою на алгоритмах підсиленого навчання, зокрема Q-learning, для підвищення продуктивності й ефективності вирощування мікрозелені в сіті-фермерстві.***

М е т о д и . ***Для прийняття обґрунтованих рішень у системі контролю клімату досліджено алгоритм навчання з підкріпленням (Q-learning), який складається з таких етапів: визначення різних кліматичних станів системи, вибір дії, яку слід виконати, на основі поточного стану системи й оцінки корисності, яку розраховують на основі рівняння Беллмана. Розроблено та реалізовано модель керування мікрокліматом, яка використовує алгоритм Q-learning для оптимізації параметрів клімату. Методика дослідження включала моделювання різних умов середовища, навчання моделі на основі зібраних даних та експериментального тестування в реальних умовах сіті-фермерства.***

Р е з у л ь т а т и . ***Експериментальне моделювання з використанням мови програмування Python із бібліотеками TensorFlow, PyTorch та scikit-learn, підтвердили ефективність застосування алгоритму Q-learning у системі контролю клімату для підвищення продуктивності й ефективності вирощування мікрозелені. Щоб переконатися, що система досягла бажаного стану, використовують такі стратегії, як моніторинг реальних значень параметрів за допомогою IoT-датчиків системи контролю клімату, аналіз отриманих значень Q-таблиці та встановлення критеріїв зупинки навчання. Результати роботи програми передають актуаторам через мережу передачі даних Wi-Fi за допомогою мікроконтролера ESP8266, який використовують як модуль Wi-Fi для мікроконтролера Arduino.***

В и с н о в к и . ***Застосування системи контролю клімату з алгоритмом Q-learning у сіті-фермерстві сприяє досягненню більшої продуктивності, ефективності та стабільності вирощування рослин, що відображається на покращенні результатів вирощування рослин.***

К л ю ч о в і с л о в а : ***розподілені системи, технології IoT, хмарні обчислення, Q-learning, моніторинг.***

Вступ

Сіті-фермерство, або міське землеробство, має значну актуальність у сучасному світі з огляду на кілька важливих факторів. За даними ООН, до 2050 р. більше 68 % світового населення мешкатиме в урбанізованих зонах. Урбаністичне фермерство здатне надати жителям міст доступ до свіжих продуктів, знижуючи потребу в імпорті з віддалених регіонів. Вирощування продуктів харчування на міських фермах може зменшити вуглецевий слід, пов'язаний із транспортуванням їжі на великі відстані. Це сприяє зменшенню емісій шкідливих газів і покращенню екологічної сталості.

У сучасному контексті ефективність сіті-фермерства значною мірою залежить від упровадження таких технологій, як розподілені системи, технології IoT (інтернет речей), хмарні обчислення та методи штучного інтелекту, як-от Q-learning. Розподілені системи дозволяють керувати різними аспектами міських ферм на віддалених платформах, забезпечуючи взаємодію між численними датчиками та пристроями. Це особливо корисно для автоматизованого моніторингу і контролю параметрів навколишнього середовища (напр., вологості або освітлення), що впливають на ріст рослин.

IoT-технології відіграють ключову роль у моніторингу й оптимізації міських ферм. Датчики можуть збирати інформацію про стан ґрунту, вологість, температуру й інші параметри в режимі реального часу, що дозволяє автоматично регулювати умови вирощування. Ці дані передаються до хмарних обчислювальних систем, де здійснюється їх оброблення для прийняття рішень і керування фермерськими процесами.

Застосування Q-learning дозволяє оптимізувати процеси керування міськими фермами за допомогою автоматичного навчання систем на основі попереднього досвіду та прогнозування найкращих умов для вирощування продуктів. Це знижує необхідність ручного втручання та підвищує ефективність процесів.

Розвиток сіті-фермерства створює нові можливості для підприємців і місцевих фермерів, забезпечуючи робочі місця та стимулюючи економічний розвиток у міських областях. Вирощування продуктів у міських умовах дозволяє контролювати умови вирощування, якість ґрунту, використання пестицидів та інших хімічних речовин, що може привести до покращення якості продуктів. Моніторинг за допомогою IoT допомагає відстежувати ці параметри та гарантувати стабільність вирощування.

Отже, для ефективного розроблення й експлуатації сіті-фермерських систем необхідно потрібно подолати не тільки звичні труднощі, такі як обмежений простір чи фінансування, але й упровадити сучасні технологічні рішення, зокрема розподілені системи, IoT, хмарні обчислення й алгоритми машинного навчання для моніторингу й оптимізації процесів.

Постановка проблеми розроблення й експлуатації сіті-фермерства полягає у забезпеченні оптимального керування температурою та вологістю, що може допомогти ефективніше використовувати обмежений простір, збільшуючи врожайність на одиницю площі. Системи контролю клімату, які дозволяють створювати оптимальні умови для рослин, забезпечують збільшення виробництва в умовах обмеженого простору.

Упровадження передових технологій контролю клімату потребує розроблення та створення відповідної інфраструктури для їхньої ефективної роботи. Нові технології контролю клімату можуть вимагати підтримки й інтеграції з існуючими системами керування сіті-фермерства.

Мета дослідження – прийняття обґрунтованих рішень у системі контролю клімату, заснованою на алгоритмах підсиленого навчання, зокрема і Q-learning, для підвищення продуктивності й ефективності вирощування мікрозелені в сіті-фермерстві.

Завдання дослідження:

- розробити модель для керування параметрами мікроклімату (температура, вологість, освітлення) на основі алгоритмів підсиленого навчання;
- реалізувати алгоритм Q-learning для автоматизації керування кліматом у сіті-фермі;
- провести експериментальну перевірку ефективності запропонованої системи на основі реальних даних щодо росту мікрозелені;
- оцінити вплив оптимізованого керування мікрокліматом на врожайність та ефективність використання простору.

Об'єктом дослідження є процеси контролю мікроклімату в системах сіті-фермерства, зокрема і керування температурою та вологістю в умовах обмеженого простору, з метою підвищення ефективності виробництва мікрозелені через застосування інтелектуальних алгоритмів, таких як Q-learning.

Наукова цінність дослідження полягає в інтеграції алгоритмів підсиленого навчання, таких як Q-learning, у системи контролю мікроклімату, що дозволяє динамічне адаптувати параметри навколишнього середовища для оптимального росту мікрозелені. Використання Q-learning забезпечує можливість самонавчання системи на основі історичних даних і постійного вдосконалення моделей керування температурою, вологістю та освітленням. Такий підхід є новаторським у сфері автоматизації сіті-фермерства і дозволяє підвищити ефективність використання обмеженого простору та ресурсів.

Огляд літератури. Концепція мікрозелені є відносно новою і з'явилася в Сан-Франциско наприкінці 1980-х рр. Вирощування мікрозелені швидко набуває популярності, особливо в умовах міського середовища, завдяки обмеженій потребі у ресурсах і можливості вирощування протягом всього року (Enssle, 2020). Проте для оптимізації процесу виробництва мікрозелені критичним є використання передових технологій, зокрема й інтелектуальних і розподілених систем керування та моніторингу (Zhou, 2020), (Ngo, Le-Khac, & Kechadi, 2018).

Міські ферми, які застосовують IoT, дозволяють здійснювати моніторинг параметрів мікроклімату в реальному часі, використовуючи датчики, що збирають інформацію про температуру, вологість, освітлення та інші важливі показники (Lodge, 2019). Дані, отримані з цих сенсорів, обробляються хмарними обчислювальними системами (Ashokkumar, Chowdary, & Sree, 2019) та за допомогою машинного навчання (Ashcraft, & Karra, 2021), що дозволяє вчасно виявляти відхилення та вносити корективи в умови вирощування.

Інтелектуальні алгоритми, такі як Q-learning, можна використовувати для аналізу та прогнозування параметрів клімату, що сприятиме максимальній ефективності використання ресурсів (Ali et al., 2024). Крім того, впровадження автоматизованих систем керування з використанням роботів підвищить точність ішвидкість реагування на зміни в умовах виробництва, зменшивши людський фактор і ризики помилок (Chougule, & Mashalkar, 2022).

Для оптимізації теплових умов також використовують теплові моделі та програмне забезпечення для моделювання клімату в теплицях, що сприяє підвищенню врожайності та якості продукції (Choab et al., 2019). Системи на основі штучного інтелекту здатні обробляти великі обсяги даних і передбачати зміни в умовах, що дозволяють забезпечити стабільність виробництва (Alibabaei, Gaspar, & Lima, 2021). Отже, розроблення та експлуатація сіті-фермерського підприємства, яке використовує передові системи контролю клімату, дозволить оптимізувати та стабілізувати виробництво, підвищить економічну ефективність і збереже ресурси.

Хоча системи контролю клімату мають багато переваг, їхнє використання також може супроводжуватися деякими недоліками, багато з них можуть бути подолані завдяки використанню сучасних інтелектуальних технологій:

- встановлення сенсорів та пристроїв IoT у різних ділянках сіті-фермерського господарства дозволяє моніторити параметри клімату та керувати ними в реальному часі (Axak et al., 2020). Це допоможе швидко виявляти проблеми та реагувати на них, зменшуючи ризики витрат і втрат врожаю (Axak, Korablyov, & Ushakov, 2020);
- системи на основі штучного інтелекту можуть аналізувати великі обсяги даних із сенсорів і прогнозувати оптимальні параметри клімату для підтримки росту рослин. Це допоможе оптимізувати використання ресурсів і забезпечувати стабільність виробництва;
- використання автоматизованих систем і роботів для налагодження та керування системами контролю клімату може підвищити ефективність і точність процесів, зменшуючи залежність від людського фактора та ризики помилок;
- технологія блокчейн може використовуватися для створення децентралізованих систем керування даними про параметри клімату та виробництва, що забезпечує надійність і прозорість обміну інформацією між усіма учасниками системи.

Цей огляд показав, що застосування сучасних IT-технологій у системах контролю клімату може забезпечити ефективне сільське господарство, зменшити втрати та покращити якість продукції.

Отже, сучасні технології інтелектуальних і розподілених систем, IoT та хмарні обчислення відіграють ключову роль у покращенні процесів керування міським фермерством, що забезпечує ефективне використання ресурсів і стабільне виробництво.

Методи

Мікрозелень вирощується з використанням різних матеріалів і може вироблятися в різних середовищах. Основні матеріали, задіяні у вирощуванні мікрозелені, містять насіння, середовище для вирощування, світло, воду та лотки. Кожен сорт мікрозелені має різну тривалість циклу росту. Мікрозелень можна вирощувати за природного або штучного освітлення. Ідеальна температура мікрозелені залежить від сорту, але зазвичай мікрозелень можна вирощувати за

ISSN 2788-6603

температури 24 °C. Потік повітря важливий, щоб запобігти накопиченню вологи та появі цвілі на мікрозелені. Вентилятор сприяє циркуляції повітря навколо мікрозеленої зони. Коріння мікрозелені та середовище для вирощування варто підтримувати у вологому стані, щоб максимізувати проростання насіння.

Вирощування мікрозелені відбувається у закритому приміщенні, де контроль за кліматом є ключовим фактором. Пропонована комп'ютеризована система контролю клімату дозволяє керувати критичними параметрами середовища вирощування, такими як температура, вологість, освітлення та циркуляція повітря, що є ключовими факторами для стабільного росту мікрозелені в закритих приміщеннях. Ця система базується на застосуванні датчиків для моніторингу параметрів клімату та використовує математичні моделі, які визначають оптимальні значення мікроклімату.

Комп'ютеризована система контролю клімату використовує алгоритми для автоматизованого регулювання параметрів, таких як температура та вологість, забезпечуючи їх оптимальність для рослин на різних етапах росту. Наприклад, система може автоматично налаштовувати потужність обігрівачів або кондиціонерів для підтримання температури на рівні 24 °C, а також контролювати вологість і циркуляцію повітря, щоб запобігти розвитку плісняви. Крім цього, систему оснащено блоком штучного освітлення, яке працює в тандемі з датчиками освітленості для забезпечення стабільного росту рослин, навіть в умовах нестачі природного світла. Усі дані з датчиків аналізуються за допомогою алгоритмів машинного навчання, що дозволяє оптимізувати роботу системи й адаптувати її до змінних умов середовища.

Завдяки впровадженню таких технологій, система зменшує ризики, пов'язані з неефективним використанням ресурсів і забезпечує стабільний рівень продуктивності сіті-ферм, максимізуючи врожайність на одиницю площі. Структуру комп'ютеризованої системи контролю клімату для сіті-фермерства наведено на рис. 1.

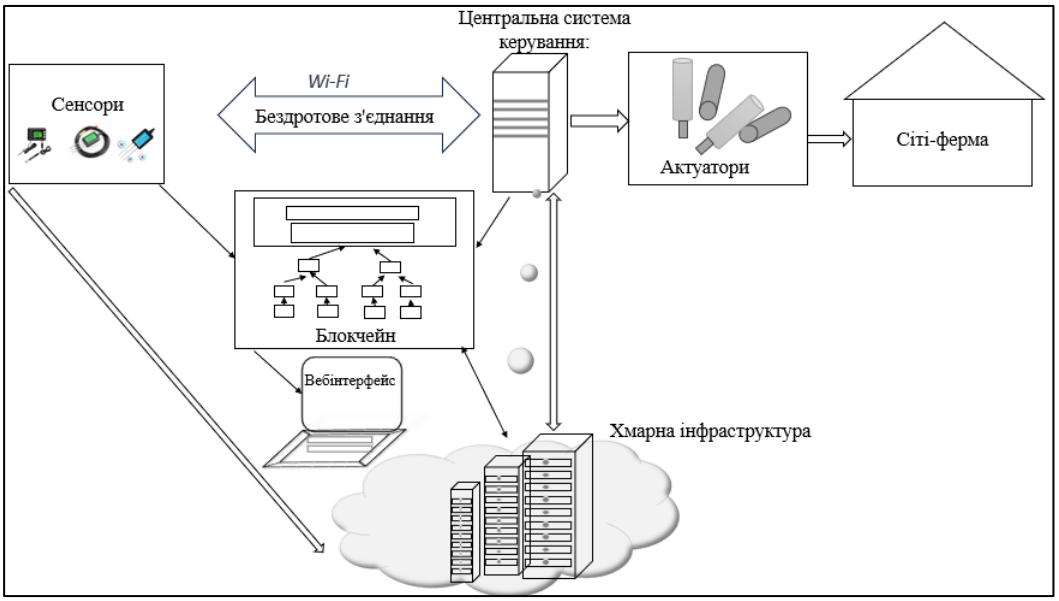


Рис. 1. Структура комп'ютеризованої системи контролю клімату для сіті-фермерства

Сенсори надсилають дані про параметри клімату (температура, вологість, освітленість тощо) до центральної системи через бездротове з'єднання: Wi-Fi або Bluetooth. Дані, що надходять від сенсорів про параметри клімату, можуть бути записані у блокчейн. Блокчейн використовують у системі для забезпечення надійного зберігання даних про параметри клімату, що надходять від сенсорів, та результатів управлінських дій. Однією з основних причин використання блокчейну є гарантія цілісності й автентичності інформації, яка важлива для точного моніторингу умов вирощування та прийняття правильних рішень. У розподіленому реєстрі блокчейну всі записи незмінні, що робить систему стійкою до зловживань або маніпуляцій даними. Це особливо актуально для гарантування того, що всі зміни в параметрах мікроклімату є прозорими і користувачі можуть довіряти інформації, яку вони отримують. Крім того, блокчейн дозволяє створювати автоматизовані процеси через смарт-контракти. Наприклад, як тільки певні кліматичні показники досягають визначених значень, система може автоматично активувати відповідні дії (вмикати обігрів або зрошення), що забезпечує ефективність і швидкість реагування. Використання блокчейну також спрощує аудит даних про виробництво, оскільки всі операції фіксуються у відкритому, але захищеному середовищі, доступному для перевірки. Блокчейн у цій системі не лише підвищує рівень безпеки та прозорості, а й дозволяє автоматизувати процеси керування на основі достовірних даних.

Центральна система отримує дані від сенсорів та аналізує їх, використовуючи алгоритми машинного навчання для прийняття рішень. Вона передає команди до актуаторів на основі аналізу даних, що включають керування системами обігріву, кондиціонування повітря, поливу й освітлення.

Актуатори приймають команди від центральної системи й автоматично регулюють параметри клімату відповідно до заданих значень. Наприклад, актуатори можуть вмикати або вимикати системи обігріву, системи поливу або регулювати яскравість освітлення. Керування актуаторами й автоматизацією відбувається на основі даних, зафіксованих у блокчейні, що гарантує їхню незмінність і безпеку. Розумні контракти, наприклад, автоматично виконують дії, коли досягаються певні умови в кліматичних даних, записаних у блокчейні.

Хмарна інфраструктура забезпечує зберігання й оброблення великих обсягів даних, які надходять від сенсорів і центральної системи. Вона також може забезпечувати можливість резервного копіювання даних і надійність роботи системи.

Блокчейн-технологію використовують для забезпечення безпеки та цілісності даних, а також для створення децентралізованої системи керування даними про параметри клімату та виробництва. Інформація про статус виробництва, керування даними й інші важливі параметри захищені за допомогою технології блокчейн.

Користувачі можуть взаємодіяти із системою через мобільний додаток або вебінтерфейс, що дозволяє моніторити стан сіті-фермерства, керувати параметрами клімату й отримувати повідомлення про події та статус виробництва. Дані, що доступні для користувачів через мобільний додаток або вебінтерфейс, можуть бути підтверджені та перевірені за допомогою блокчейну. Це дозволяє забезпечити користувачам достовірність і цілісність інформації, яку вони отримують.

Для прийняття рішень у системі контролю клімату пропонується *алгоритм навчання з підкріпленням* (Q-learning), який складається з таких етапів.

Етап 1. Визначення різних станів системи – температури, вологості, освітлення тощо. Кожен стан представлено у вигляді вектора ознак.

Етап 2. Вибір дії, яку потрібно виконати, на основі поточного стану системи та оцінки корисності (Q-значення) кожної доступної дії:

Поточний стан системи:	*Температура в теплиці вища за оптимальне значення для вирощування рослин.*
Доступні дії:	*Дія 1. Збільшити час роботи люмінесцентних ламп для підвищення рівня освітлення.*
	Дія 2. Зменшити використання системи обігріву або припинити його зовсім.
	Дія 3. Збільшити вологість у теплиці.
Оцінка корисності (Q-значення):	*Q1 = 0,8 (для дії 1).*
	Q2 = 0,5 (для дії 2).
	Q3 = 0,6 (для дії 3).
Вибір дії:	*Обираємо дію з найвищим Q-значенням: дія 1 (збільшення часу роботи люмінесцентних ламп).*

Отже, система автоматично виконає дію 1, щоб знизити температуру в теплиці та забезпечити оптимальні умови для росту рослин.

Етап 3. Взаємодія із середовищем, щодо регулювання параметрів клімату згідно з обраною дією:

Отримання поточних значень параметрів клімату (температура, вологість тощо).

Аналіз поточного стану.

Виконання обраної дії (напр., збільшення часу роботи вентиляційної системи).

Моніторинг результатів.

Прийняття подальших рішень.

Процес взаємодії із середовищем для регулювання параметрів клімату повторюють періодично для підтримання оптимальних умов у теплиці.

Етап 4. Отримання нагороди, яка відображає, наскільки добре ця дія відповідає поточному стану системи. Система аналізує отримані дані після виконання дії, включаючи зміни параметрів клімату й ефективність керування. На основі результатів аналізу формується метрика kpi, що відображає ефективність виконаної дії, якщо ми, наприклад, хочемо оцінити, наскільки добре дія відповідає поточному стану системи (1). Наприклад, це може бути різниця між поточною температурою і оптимальною температурою, або кількість часу, протягом якого параметри клімату перебували в оптимальних межах.

$$\mathrm{kpi}= (\mathrm{S_{opt}} - \mathrm{S_{cur}})/\mathrm{S_{opt}}\ , \tag{1}$$

де $\mathrm{S_{opt}}$ – оптимальний стан системи – це значення параметра клімату, яке вважається ідеальним для вирощування рослин (напр., оптимальна температура або вологість); $\mathrm{S_{cur}}$ – поточний стан системи – це фактичне значення параметра клімату після виконання дії.

Функція нагороди враховує базові параметри і визначає значення нагороди для кожної виконаної дії (2).

$$\mathrm{Q} = \mathrm{w_1 p_1} + \mathrm{w_2 p_2} + \cdots + \mathrm{w_n p_n}, \tag{2}$$

де $\mathrm{w_1}, \mathrm{w_2}, \ldots, \mathrm{w_n}$ – вагові коефіцієнти, які відображають важливість кожного параметра $\mathrm{p_i}$, такого як ефективність дії, ступінь відповідності поточного стану системи до оптимального, або будь-які інші критерії, що характеризують успішність виконаної дії.

Після виконання дії в середовищі з'являється зворотний зв'язок у вигляді нагороди за цю дію. Ця нагорода може бути збережена або акумульована з попередніми нагородами. Наприклад, якщо намагатися максимізувати загальну нагороду протягом певного періоду часу, то може зберігатися поточна сума нагороди та додаватися до неї отримана нагорода за кожну дію.

ISSN 2788-6603

Після отримання нагороди за виконану дію, оновлюється відповідне Q-значення в Q-таблиці. Цей процес оновлення здійснюється за допомогою формули оновлення Q-значень в алгоритмі Q-learning (3):

$$Q^{new}(s,a) = Q^{old}(s,a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s',a')), \quad (3)$$

де $Q(s,a)$ – поточне значення Q-функції для стану s та дії a; α – коефіцієнт навчання (learning rate), який визначає, наскільки значущим є нова інформація (нагорода); r – нагорода, яку агент отримав за виконану дію; γ – дисконтний фактор (discount factor), який визначає, наскільки значущими є майбутні нагороди; s' – наступний стан, в який перейшов агент після виконання дії a; a' – можлива дія в наступному стані s'.

Отриману нагороду використовують для оновлення Q-значень, які потім застосовує агент для вибору наступної дії в майбутній взаємодії із середовищем. Цей процес дозволяє навчатися і вдосконалювати свої стратегії, щоб досягти максимальної нагороди в майбутньому.

Етап 5. Оновлення Q-значень для поточного стану й обраної дії, з використанням формули оновлення Q-значень (4):

$$Q^{new}(s,a) = (1-\alpha) \cdot Q^{old}(s,a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s',a')). \quad (4)$$

Етап 6. Продовження навчання протягом багатьох ітерацій, з оновленням Q-значення на кожному кроці з метою максимізації загальної нагороди в майбутньому.

Цей процес навчання дозволяє системі контролю клімату вибирати оптимальні дії для забезпечення оптимальних умов для росту рослин у системі сіті-фермерства. Із часом система навчається оптимальних стратегій керування кліматом із метою максимізації врожаю або інших бажаних цілей.

Результати

Для реалізації алгоритму навчання з підкріпленням (Q-learning) в системі контролю клімату використовують Python – це одна з найпопулярніших мов програмування для реалізації алгоритмів машинного навчання, включаючи Q-learning. У Python існують бібліотеки, такі як TensorFlow, PyTorch, scikit-learn, які містять реалізації Q-learning.

Створено клас QLearningAgent, який містить методи для вибору дії та оновлення таблиці Q-значень. Потім ми створюємо екземпляр цього агента й використовуємо його для вибору дій та оновлення Q-таблиці в кожній ітерації (рис. 2).

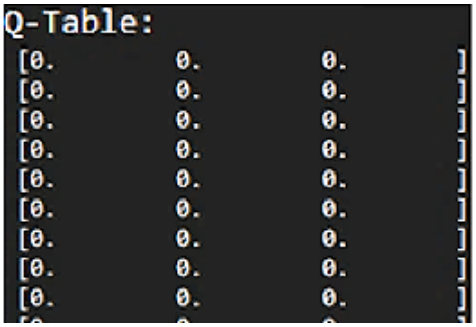


Рис. 2. Приклад вмісту Q-таблиці на етапі ініціалізації

Після кожного оновлення Q-таблиці, ми можемо бачити, як змінюються значення Q-функції для кожного стану та дії. Найбільше значення Q-функції для кожного стану вказує на те, яка дія найвигідніша. Високі значення Q-функції свідчать про перевагу певних дій у поточному стані системи, що сприяють досягненню оптимальних умов мікроклімату.

Наприклад, якщо поточний стан системи представлений вектором s_{cur} = [19, 70, 3500], де зазначені відповідні значення температури, вологості й інтенсивності освітлення, то алгоритм Q-learning відреагує таким способом.

Оскільки вказана температура (19 °C) нижча за оптимальну, алгоритм Q-learning обирає дії, спрямовані на підвищення температури, коли поточний стан системи потребує підвищення цього параметра для досягнення оптимальних умов для росту рослин, наприклад, увімкнувши систему опалення в теплиці. Вологість, яка вказана у векторі стану (70 %), незначно перевищує оптимальну вологість (60 %). Однак ця різниця не є критичною, тому алгоритм може залишити вологість без змін або, залежно від його налаштувань, може виконати додаткові заходи для підтримки оптимального рівня вологості. Вказана інтенсивність освітлення (3500 люкс) нижча за оптимальний рівень (5000 люкс). Тому алгоритм може вирішити збільшити інтенсивність світла, увімкнувши більше світлодіодних ламп або збільшивши час їхньої роботи. Нульові значення в таблиці свідчать про те, що поки не здійснено жодну дію та не отримано нагороду.

Навчання алгоритму Q-learning відбувається за допомогою таких кроків.

1. *Ініціалізація Q-таблиці нульовими значеннями.*
2. *Вибір дії на основі вивчених значень Q-таблиці.*
3. *Виконання дії у поточному стані та вибір нагороди на основі взаємодії із середовищем.*
4. *Оновлення Q-значень.*
5. *Повторення кожного кроку до досягнення критерію зупинки.*
6. *Експлуатація навченої стратегії.*

Щоб прийняти рішення для вектора стану системи заданого як s_{cur} = [19, 70, 3500], ми використовуємо навчену Q-таблицю, щоб визначити, яку дію вибрати. Спочатку порівнюємо поточний стан системи з оптимальним станом,

тобто s_{opt} = [24, 60, 5000]. Для кожного параметра визначаємо, чи потрібно збільшувати, зменшувати або залишати його незмінним, та визначаємо дію, яку потрібно вибрати на основі значень Q-таблиці. У Q-таблиці кожен стан подано рядком, а кожну дію – стовпцем. В цьому прикладі є три можливі дії (збільшення, зменшення та залишати без змін) для трьох параметрів (температура, вологість та інтенсивність освітлення), кожен елемент [i, j] представляє Q-значення для дії j у стані i (табл. 1).

Таблиця 1

Q-таблиця для системи з трьома параметрами та трьома можливими діями для кожного параметра

State	Action 1 (Decrease)	Action 2 (Unchanged)	Action 3 (Increase)
1 Low Temp, Low Humidity, Low Light	Q_{11}	Q_{12}	Q_{13}
2 Low Temp, Low Humidity, Unchanged Light	Q_{21}	Q_{22}	Q_{23}
3 Low Temp, Low Humidity, High Light	Q_{31}	Q_{32}	Q_{33}
4 Low Temp, Unchanged Humidity, Low Light	Q_{41}	Q_{42}	Q_{43}
5 Low Temp, Unchanged Humidity, Unchanged Light	Q_{51}	Q_{52}	Q_{53}
6 Low Temp, Unchanged Humidity, High Light	Q_{61}	Q_{62}	Q_{63}
7 Low Temp, High Humidity, Low Light	Q_{71}	Q_{72}	Q_{73}
8 Low Temp, High Humidity, Unchanged Light	Q_{81}	Q_{82}	Q_{83}
9 Low Temp, High Humidity, High Light	Q_{91}	Q_{92}	Q_{93}
10 Unchanged Temp, Low Humidity, Low Light	$Q_{10\,1}$	$Q_{10\,2}$	$Q_{10\,3}$
11 Unchanged Temp, Low Humidity, Unchanged Light	$Q_{11\,1}$	$Q_{11\,2}$	$Q_{11\,3}$
12 Unchanged Temp, Low Humidity, High Light	$Q_{12\,1}$	$Q_{12\,2}$	$Q_{12\,3}$
13 Unchanged Temp, Unchanged Humidity, Low Light	$Q_{13\,1}$	$Q_{13\,2}$	$Q_{13\,3}$
14 Unchanged Temp, Unchanged Humidity, Unchanged Light	$Q_{14\,1}$	$Q_{14\,2}$	$Q_{14\,3}$
15 Unchanged Temp, Unchanged Humidity, High Light	$Q_{15\,1}$	$Q_{15\,2}$	$Q_{15\,3}$
16 Unchanged Temp, High Humidity, Low Light	$Q_{16\,1}$	$Q_{16\,2}$	$Q_{16\,3}$
17 Unchanged Temp, High Humidity, Unchanged Light	$Q_{17\,1}$	$Q_{17\,2}$	$Q_{17\,3}$
18 Unchanged Temp, High Humidity, High Light	$Q_{18\,1}$	$Q_{18\,2}$	$Q_{18\,3}$
19 High Temp, Low Humidity, Low Light	$Q_{19\,1}$	$Q_{19\,2}$	$Q_{19\,3}$
20 High Temp, Low Humidity, Unchanged Light	$Q_{20\,1}$	$Q_{20\,2}$	$Q_{20\,3}$
21 High Temp, Low Humidity, High Light	$Q_{21\,1}$	$Q_{21\,2}$	$Q_{21\,3}$
22 High Temp, Unchanged Humidity, Low Light	$Q_{22\,1}$	$Q_{22\,2}$	$Q_{22\,3}$
23 High Temp, Unchanged Humidity, Unchanged Light	$Q_{23\,1}$	$Q_{23\,2}$	$Q_{23\,3}$
24 High Temp, Unchanged Humidity, High Light	$Q_{24\,1}$	$Q_{24\,2}$	$Q_{24\,3}$
25 High Temp, High Humidity, Low Light	$Q_{25\,1}$	$Q_{25\,2}$	$Q_{25\,3}$
26 High Temp, High Humidity, Unchanged Light	$Q_{26\,1}$	$Q_{26\,2}$	$Q_{26\,3}$
27 High Temp, High Humidity, High Light	$Q_{27\,1}$	$Q_{27\,2}$	$Q_{27\,3}$

Оцінити ефективність Q-алгоритму можна за допомогою середнього значення нагороди (Average Reward). Ця статистична метрика використовується для оцінювання продуктивності алгоритму Q-навчання і вимірює середню нагороду, отриману агентом під час взаємодії із середовищем протягом певного періоду часу або кількості епох. Середнє значення нагороди обчислюють як суму всіх отриманих нагород протягом епізодів, поділену на загальну кількість епізодів (5):

$$\text{Average Reward} = \frac{\sum_{i=1}^{N} R_i}{N}, \tag{5}$$

де R_i – нагорода, отримана в i-му епізоді, обчислюється як сума нагород за кожен крок у межах епізоду: $R_i = \sum_{t=1}^{T} r_t$, з кількістю кроків T; N – загальна кількість епізодів.

Ця метрика дозволяє оцінити ефективність алгоритму. Якщо середнє значення нагороди збільшується із часом і наближається до максимально можливої нагороди в середовищі, це може свідчити про успішність навчання.

На рис. 3 показано результати, що представляють оновлені значення Q-таблиці після виконання алгоритму Q-learning для кількох можливих станів системи контролю клімату.

Кожен рядок таблиці відповідає певному стану системи, який визначається значеннями температури, вологості й інтенсивності освітлення. Наприклад, State [19, 70, 3500] вказує на температуру 19 °C, вологість 70 % та інтенсивність освітлення 3500 люкс.

Кожен стовпець у рядку відповідає можливій дії, яку можна виконати в даному стані. Значення у кожному стовпці вказує на очікувану користь (Q-значення) від вибору цієї дії у цьому стані. Чим більше значення Q, тим більшою вважають користь від цієї дії.

ISSN 2788-6603

Вказані результати свідчать про те, які дії є найвигіднішими для кожного стану системи контролю клімату (рис. 4).

```
Updated Q-table:
State [19, 70, 3500]: [0.00400059 0.00022393 0.00139468]
State [19, 70, 5000]: [0.10177931 0.33295297 0.08988007]
State [19, 70, 6500]: [0.0039214  0.00111762 0.001602  ]
State [19, 71, 3500]: [0.00058998 0.00397421 0.00107031]
State [19, 71, 5000]: [0.32373891 0.07093993 0.03967545]
State [19, 71, 6500]: [0.00394546 0.0010813  0.00102636]
State [19, 72, 3500]: [0.00399532 0.00123542 0.00059964]
State [19, 72, 5000]: [0.30560804 0.06571434 0.04555891]
State [19, 72, 6500]: [0.00404674 0.00072275 0.00056592]
State [20, 70, 3500]: [0.00402911 0.00069436 0.00067158]
State [20, 70, 5000]: [0.36408993 0.08154069 0.        ]
State [20, 70, 6500]: [0.00408011 0.00056451 0.00069816]
State [20, 71, 3500]: [0.00400059 0.00065257 0.00105252]
State [20, 71, 5000]: [0.33295297 0.10713358 0.07394444]
State [20, 71, 6500]: [0.0039214  0.00081124 0.00143449]
State [20, 72, 3500]: [0.00412466 0.00036807 0.00011707]
State [20, 72, 5000]: [0.31962577 0.07806135 0.04194357]
State [20, 72, 6500]: [0.00402381 0.00133344 0.00035919]
State [21, 70, 3500]: [0.0040828  0.00027964 0.00048318]
State [21, 70, 5000]: [0.38443995 0.06160556 0.04187721]
State [21, 70, 6500]: [0.00405741 0.00059012 0.00070884]
State [21, 71, 3500]: [0.0041301 0.        0.0006573]
State [21, 71, 5000]: [0.35953887 0.06642935 0.10167183]
State [21, 71, 6500]: [0.00400322 0.00099791 0.00068689]
State [21, 72, 3500]: [0.00389447 0.00192507 0.00078557]
State [21, 72, 5000]: [0.33738276 0.08290635 0.04949187]
State [21, 72, 6500]: [9.49334355e-05 4.07742346e-03 8.81331386e-04]
```

Рис. 3. Приклад вмісту Q-таблиці після виконання алгоритму Q-learning

```
Final Q-table:
[[0.00658762 0.00658762 0.00658762]
 [0.00658762 0.00658762 0.00658762]
 [0.00658762 0.00658762 0.00658762]]
```

Рис. 4. Оновлені значення Q-таблиці після завершення навчання

Результати в табл. 1 показують оновлені значення Q-таблиці під час навчання за допомогою Q-learning алгоритму. Для стану s_{cur} = [19, 70, 3500] значення Q-таблиці [0.00658762 0.00658762 0.00658762] показують корисність вибору кожної з трьох можливих дій. Тут значення Average Reward = 0.001976284584980403 визначає середнє значення нагороди, яке отримують за кожен епізод під час навчання алгоритму Q-learning. Це значення обчислюють як середнє арифметичне з усіх нагород за епізоди. У цьому випадку, вибіркова нагорода за кожен епізод обчислюється як обернена відстань між поточним станом системи й оптимальним станом [24, 60, 5000], а потім підсумовується для кожного епізоду. На початку навчання, якщо середнє значення нагороди залишається низьким або стабільним, це може означати, що алгоритм не зміг навчитися ефективно пристосовуватися до середовища і потребує подальшого налагодження.

Щоб досягнути стабільного стану, де значення параметрів температури, вологості й інтенсивності освітлення набули оптимальних значень (s_{opt} = [24, 60, 5000]), оцінюємо стабільність за допомогою функції `check_stable_state`, тобто порівнюємо значення параметрів стану моделі з оптимальними значеннями. Перевіряємо середнє значення Q-таблиці. Після того, як програма виконала 5000 епох навчання, був досягнутий стабільний стан і Average Reward набув значення більше 0,9, то вважаємо, що досягнуто бажаного стабільного стану, тобто значення параметрів досягли оптимальних значень, навчання зупиняється.

Ці значення оновлюють у процесі навчання з урахуванням отриманих нагород і максимального очікуваного Q-значення для наступного стану, що допомагає вибрати найоптимальнішу дію для кожного стану системи контролю клімату.

Щоб переконатися, що система досягла бажаного стану, використовують кілька стратегій:

1. Моніторинг реальних значень параметрів. Якщо контрольовані параметри зближаються або залишаються в межах бажаних значень (напр., температура наближається до 24 °C, вологість до 60 %, інтенсивність світла до 5000 люкс), то це свідчить про досягнення стабільного стану.
2. Аналіз значень Q-таблиці. Після навчання моделі і вдосконалення Q-таблиці аналізують значення Q-функції для кожного можливого стану та дії. Якщо значення Q-функції для кожної дії в кожному стані наближається до максимального значення, то це свідчить про досягнення стабільного стану.
3. Встановлення критеріїв зупинки навчання. В програмі встановлено критерії зупинки навчання, коли значення Q-таблиці досягли певного діапазону або коли зміни параметрів системи вже не дуже великі. Коли ці критерії виконуються, навчання вважають завершеним, і систему можна вважати стабільною.

Після аналізу Q-таблиці можна виявити можливості для подальшого вдосконалення алгоритму. Наприклад, якщо деякі стани мають низькі значення Q-функції, може виникнути потреба у додатковому навчанні для виявлення оптимальних дій у цих станах.

Результати роботи програми передають актуаторам через мережу передачі даних Wi-Fi за допомогою мікроконтролера ESP8266, який використовують як Wi-Fi-модуль для мікроконтролера Arduino. Мікроконтролер ESP8266 налаштовується як вебсервер, який слухає запити на своїй локальній IP-адресі через порт 80. Arduino, що підключений до мережі Wi-Fi, відправляє GET-запит на IP-адресу ESP8266, отримує результати роботи програми з мікроконтролера ESP8266 і може виконати відповідні дії з актуаторами.

Дискусія і висновки

1. *Вплив системи контролю клімату на вирощування рослин*. Метою реалізації системи контролю клімату сіті-ферми є максимізація загальної швидкості росту мікрозелені за допомогою прийняття обґрунтованих рішень щодо регулювання кліматичних факторів (температура, вологість, освітлення) (Nikolaou et al., 2019) та характеристиками врожаю (висота, колір, вміст вологи, площа листя). Оптимальна температура та день збирання мікрозелені для отримання максимального врожаю становить від 24 до 28 °C, а найкращий період їх збирання – це 6–13-й день, відносна вологість від 65 % до 75 % (Dhaka et al., 2023). Кількість світла і якість впливають на ріст і фізіологію рослин і взаємодіють з іншими параметрами навколишнього середовища й факторами культивування, визначаючи поведінку рослин. Світло не лише забезпечує енергію для фотосинтезу, але й регулює розвиток, формування та метаболізм рослин у складному явищі фотоморфогенезу, керованого світлом кольорів (Paradiso, & Proietti, 2022).

2. *Застосування алгоритмів прийняття рішень*. Q-learning є одним з алгоритмів із підсиленого навчання, який використовують для прийняття рішень в умовах невизначеності та змінюваних середовищ. У контексті системи контролю клімату для сіті-фермерства, Q-learning може бути використаний для розроблення оптимальної стратегії керування параметрами клімату, такими як температура, вологість, освітлення тощо, з метою максимізації врожаю або зниження споживання енергії. Також Q-learning застосовують для сприйняття навколишнього середовища для продовження терміну служби мереж моніторингу птахофабрик (Wu et al., 2021), для ефективного підвищення точності керування сільськогосподарським зрошенням (Zhou, 2020), для моделювання напівзакритої теплиці в Нью-Йорку, яка споживає на 61 % менше енергії (Ajagekar, & You, 2022). Використання Q-learning у системі контролю клімату для сіті-фермерства має кілька переваг порівняно з традиційними методами: a) Q-learning може адаптуватися до змін у середовищі, оскільки він навчається в реальному часі, використовуючи інформацію про винагороди, отримані після кожної дії. Це робить його ефективним для систем, які піддаються варіаціям у внутрішніх і зовнішніх умовах, що є типовим для сіті-фермерства; b) Q-learning спроможний знаходити оптимальні стратегії керування системою контролю клімату для максимізації врожаю або зменшення споживання енергії. Він може знаходити рішення, які підходять для конкретних обмежень і цілей, визначених фермером чи оператором системи; c) системи контролю клімату у сіті-фермерстві можуть бути дуже складними, з великою кількістю змінних і взаємодіючих параметрів. Алгоритм Q-learning може ефективно керувати такими складними системами, здатний до розв'язання задачі оптимізації навіть у складних і невизначених середовищах; d) Q-learning може бути ефективно використаний у системах різних розмірів. Він може пристосовуватися до різних масштабів виробництва, від невеликих ферм до великих агропромислових комплексів.

3. *Узагальнення результатів дослідження* показує, що використання системи контролю клімату з використанням алгоритму Q-learning дозволяє досягати більшої продуктивності й ефективності вирощування у сіті-фермерстві. Система контролю клімату забезпечує точне регулювання параметрів середовища, таких як температура, вологість та освітлення, що сприяє оптимальному росту рослин. Завдяки адаптивному навчанню та постійному вдосконаленню стратегій керування, система може ефективно використовувати енергію та воду, зменшуючи витрати на виробництво. Завдяки оптимізації умов середовища для росту рослин і зменшення стресу на них, система контролю клімату сприяє підвищенню врожайності та якості продукції. Алгоритм Q-learning дозволяє системі автоматично адаптуватися до змін у зовнішніх умовах, таких як зміни погоди або сезонні зміни, забезпечуючи стабільність і надійність у вирощуванні культур. За допомогою ефективного керування умовами вирощування, система допомагає знижувати втрати та ризики для фермерів.

Отже, використання системи контролю клімату з Q-learning у сіті-фермерстві сприяє досягненню більшої продуктивності, ефективності та стабільності вирощування рослин, що веде до покращення результатів господарювання та забезпечення сталого розвитку аграрного сектору.

Внесок авторів: Наталія Аксак – концептуалізація; Максим Кушнарьов – програмне забезпечення; Юрій Шеліхов – збір емпіричних даних та їхня валідація, аналіз джерел, підготовка огляду літератури.

Список використаних джерел

Ajagekar, A., & You, F. (2022). Deep Reinforcement Learning Based Automatic Control in Semi-Closed Greenhouse Systems. In L. Ricardez-Sandoval, J. Pico, J. H. Lee, J. M. Lee (Eds.), *IFAC-PapersOnLine*, 55(7), 406–411. https://doi.org/10.1016/j.ifacol.2022.07.477

Ali, N., Wahid, A., Shaw, R., & Mason, K. (2024). A Reinforcement Learning Approach to Dairy Farm Battery Management using Q Learning. In T. Dietterich, K. Apt, R. Boisvert (Eds.), *Journal of Energy Storage*, 93, 112031. https://doi.org/10.48550/arXiv.2403.09499

Alibabaei, K., Gaspar, P. D., & Lima, T. M. (2021). Crop yield estimation using deep learning based on climate big data and irrigation scheduling. In José A. Afonso, R. Barbosa, A. Bielecki (Eds.), *Energies*, 14(11), 3004. https://doi.org/10.3390/en14113004

Ashcraft, C., & Karra, K. (2021). Machine learning aided crop yield optimization. In T. Dietterich, K. Apt, R. Boisvert (Eds.), arXiv preprint arXiv:2111.00963. https://doi.org/10.48550/arXiv.2111.00963

Ashokkumar, K., Chowdary, D. D., & Sree, C. D. (2019, October). Data analysis and prediction on cloud computing for enhancing productivity in agriculture. In *IOP Conference Series: Materials Science and Engineering* (Vol. 590, No. 1, p. 012014). IOP Publishing. doi 10.1088/1757-899X/590/1/012014

Axak, N., Korablyov, M., Ushakov, M. (2020). Cloud Architecture for Remote Medical Monitoring. *IEEE Proceedings of the 15th International conference "Computer Sciences and Information Technologies" (CSIT-2020).* 1 (pp. 344–347). Zbarazh–Lviv. https:// doi:10.1109/CSIT49958.2020.9321927

Axak N., Serdiuk N., Ushakov M., Korablyov M. (2020). Development of System for Monitoring and Forecasting of Employee Health on the Enterprise. In V. Lytvyn, V. Vysotska, T. Hamon, N. Grabar, N. Sharonova, O. Cherednichenko, O. Kanishcheva (Eds.), *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems* (COLINS 2020), April 23–24, I: Main Conference (pp. 979–992), Lviv. https://ceur-ws.org/Vol-2604/paper65.pdf

Choab, N., Allouhi, A., El Maakoul, A., Kousksou, T., Saadeddine, S., & Jamil, A. (2019). Review on greenhouse microclimate and application: Design parameters, thermal modeling and simulation, climate controlling technologies. In R. Pitchumani, X. Xia (Eds.), *Solar Energy*, 191, 109–137. https://doi.org/10.1016/j.solener.2019.08.042

Chougule, M. A., & Mashalkar, A. S. (2022). A comprehensive review of agriculture irrigation using artificial intelligence for crop production. In K. Kumar, G. Kakandikar, & J. Paulo Davim (Eds.), *Computational Intelligence in Manufacturing*, 187–200. https://doi.org/10.1016/B978-0-323-91854-1.00002-9

ISSN 2788-6603

Dhaka, A. S., Dikshit, H. K., Mishra, G. P., Tontang, M. T., Meena, N. L., Kumar, R. R., Ramesh, S. V., Narwal, S., Aski, M., Thimmegowda, V., Gupta S., Nair, & R. M., Praveen, S. (2023). Evaluation of Growth Conditions, Antioxidant Potential, and Sensory Attributes of Six Diverse Microgreens Species. In Rosario Paolo Mauro (Eds.), *Agriculture*, 13(3), 676. https://doi.org/10.3390/agriculture13030676

Enssle, N. (2020). *Microgreens: Market Analysis, Growing Methods and Models.* In N. Enssle (Eds.). California State University San Marcos.

Lodge, J. (2019). Controlled Environment Agriculture: using Intelligent Systems on the next level. In A. Muñoz, J. Park (Eds.), *Agriculture and Environment Perspectives in Intelligent Systems* (pp. 1–33). IOS Press. https://doi.org/10.3233/AISE190002

Ngo, V. M., Le-Khac, N. A., & Kechadi, M. (2018). An efficient data warehouse for crop yield prediction. In T. Dietterich, K. Apt, R. Boisvert (Eds.), *arXiv preprint arXiv:1807.00035*. https://doi.org/10.48550/arXiv.1807.00035

Nikolaou, G., Neocleous, D., Katsoulas, N., & Kittas, C. (2019). Irrigation of greenhouse crops. In A. Koukounaras (Eds.), *Horticulturae*, 5(1), 7. https://doi.org/10.3390/horticulturae5010007

Paradiso, R., Proietti, S. (2022). Light-Quality Manipulation to Control Plant Growth and Photomorphogenesis in Greenhouse Horticulture: The State of the Art and the Opportunities of Modern LED Systems. In J. Ludwig-Müller (Eds.), *J. Plant Growth Regul* ,41, 742–780. https://doi.org/10.1007/s00344-021-10337-y

Wu, Z., Pan, P., Liu, J., Shi, B., Yan, M., & Zhang, H. (2021). Environmental Perception Q-Learning to Prolong the Lifetime of Poultry Farm Monitoring Networks. In V. Varadarajan (Eds.), *Electronics*, 10(23), 3024. https://doi.org/10.3390/electronics10233024

Zhou, N. (2020). Intelligent control of agricultural irrigation based on reinforcement learning. *Journal of Physics Conference Series,*1601(5), 052031. IOP Publishing. https://doi.org/10.1088/1742-6596/1601/5/052031

References

Ajagekar, A., & You, F. (2022). Deep Reinforcement Learning Based Automatic Control in Semi-Closed Greenhouse Systems. In L. Ricardez-Sandoval, J. Pico, J. H. Lee, J. M. Lee (Eds.), *IFAC-PapersOnLine*, 55(7), 406–411. https://doi.org/10.1016/j.ifacol.2022.07.477

Ali, N., Wahid, A., Shaw, R., & Mason, K. (2024). A Reinforcement Learning Approach to Dairy Farm Battery Management using Q Learning. In T. Dietterich, K. Apt, R. Boisvert (Eds.), *Journal of Energy Storage*, 93, 112031. https://doi.org/10.48550/arXiv.2403.09499

Alibabaei, K., Gaspar, P. D., & Lima, T. M. (2021). Crop yield estimation using deep learning based on climate big data and irrigation scheduling. In José A. Afonso, R. Barbosa, A. Bielecki (Eds.), *Energies*, 14(11), 3004. https://doi.org/10.3390/en14113004

Ashcraft, C., & Karra, K. (2021). Machine learning aided crop yield optimization. In T. Dietterich, K. Apt, R. Boisvert (Eds.), arXiv preprint arXiv:2111.00963. https://doi.org/10.48550/arXiv.2111.00963

Ashokkumar, K., Chowdary, D. D., & Sree, C. D. (2019, October). Data analysis and prediction on cloud computing for enhancing productivity in agriculture. In *IOP Conference Series: Materials Science and Engineering* (Vol. 590, No. 1, p. 012014). IOP Publishing. doi 10.1088/1757-899X/590/1/012014

Axak, N., Korablyov, M., Ushakov, M. (2020). Cloud Architecture for Remote Medical Monitoring. *IEEE Proceedings of the 15th International conference "Computer Sciences and Information Technologies" (CSIT-2020).* 1 (pp. 344–347). Zbarazh–Lviv. https:// doi:10.1109/CSIT49958.2020.9321927

Axak N., Serdiuk N., Ushakov M., Korablyov M. (2020). Development of System for Monitoring and Forecasting of Employee Health on the Enterprise. In V. Lytvyn, V. Vysotska, T. Hamon, N. Grabar, N. Sharonova, O. Cherednichenko, O. Kanishcheva (Eds.), *Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Systems* (COLINS 2020), April 23–24, I: Main Conference (pp. 979–992), Lviv. https://ceur-ws.org/Vol-2604/paper65.pdf

Choab, N., Allouhi, A., El Maakoul, A., Kousksou, T., Saadeddine, S., & Jamil, A. (2019). Review on greenhouse microclimate and application: Design parameters, thermal modeling and simulation, climate controlling technologies. In R. Pitchumani, X. Xia (Eds.), *Solar Energy*, 191, 109–137. https://doi.org/10.1016/j.solener.2019.08.042

Chougule, M. A., & Mashalkar, A. S. (2022). A comprehensive review of agriculture irrigation using artificial intelligence for crop production. In K. Kumar, G. Kakandikar, & J. Paulo Davim (Eds.), *Computational Intelligence in Manufacturing*, 187–200. https://doi.org/10.1016/B978-0-323-91854-1.00002-9

Dhaka, A. S., Dikshit, H. K., Mishra, G. P., Tontang, M. T., Meena, N. L., Kumar, R. R., Ramesh, S. V., Narwal, S., Aski, M., Thimmegowda, V., Gupta S., Nair, & R. M., Praveen, S. (2023). Evaluation of Growth Conditions, Antioxidant Potential, and Sensory Attributes of Six Diverse Microgreens Species. In Rosario Paolo Mauro (Eds.), *Agriculture*, 13(3), 676. https://doi.org/10.3390/agriculture13030676

Enssle, N. (2020). *Microgreens: Market Analysis, Growing Methods and Models.* In N. Enssle (Eds.). California State University San Marcos.

Lodge, J. (2019). Controlled Environment Agriculture: using Intelligent Systems on the next level. In A. Muñoz, J. Park (Eds.), *Agriculture and Environment Perspectives in Intelligent Systems* (pp. 1–33). IOS Press. https://doi.org/10.3233/AISE190002

Ngo, V. M., Le-Khac, N. A., & Kechadi, M. (2018). An efficient data warehouse for crop yield prediction. In T. Dietterich, K. Apt, R. Boisvert (Eds.), *arXiv preprint arXiv:1807.00035*. https://doi.org/10.48550/arXiv.1807.00035

Nikolaou, G., Neocleous, D., Katsoulas, N., & Kittas, C. (2019). Irrigation of greenhouse crops. In A. Koukounaras (Eds.), *Horticulturae*, 5(1), 7. https://doi.org/10.3390/horticulturae5010007

Paradiso, R., Proietti, S. (2022). Light-Quality Manipulation to Control Plant Growth and Photomorphogenesis in Greenhouse Horticulture: The State of the Art and the Opportunities of Modern LED Systems. In J. Ludwig-Müller (Eds.), *J. Plant Growth Regul* ,41, 742–780. https://doi.org/10.1007/s00344-021-10337-y

Wu, Z., Pan, P., Liu, J., Shi, B., Yan, M., & Zhang, H. (2021). Environmental Perception Q-Learning to Prolong the Lifetime of Poultry Farm Monitoring Networks. In V. Varadarajan (Eds.), *Electronics*, 10(23), 3024. https://doi.org/10.3390/electronics10233024

Zhou, N. (2020). Intelligent control of agricultural irrigation based on reinforcement learning. *Journal of Physics Conference Series,*1601(5), 052031. IOP Publishing. https://doi.org/10.1088/1742-6596/1601/5/052031

Отримано редакцією журналу / Received: 15.08.24
Прорецензовано / Revised: 24.09.24
Схвалено до друку / Accepted: 06.10.24

Natalia AXAK, DSc (Engin.), Prof.
ORCID ID: 0000-0001-8372-8432
e-mail: nataliia.axak@nure.ua
Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

Maksym KUSHNARYOV, PhD (Engin.)
ORCID ID: 0000-0002-3772-3195
e-mail: maksym.kushnarov@nure.ua
Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

Yurii SHELIKHOV, PhD Student
ORCID ID: 0009-0009-8970-6571
e-mail: yurii.shelikhov@nure.ua
Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

THE INTELLIGENT CONTROL OF THE CITY-FARM MICROCLIMATE BASED ON THE Q-LEARNING ALGORITHM

Background. *In the context of the rapid development of urban farming and the growing interest in sustainable food production, microclimate management is becoming a key aspect to achieve optimal plant cultivation. Optimum management of temperature, humidity and light can help use limited space more efficiently, increasing yield per unit area. Climate control systems that allow you to create optimal conditions for plants allow you to increase production in a limited area. The purpose of the study is to make informed decisions in the climate control system based on reinforcement learning algorithms, in particular Q-learning, to increase the productivity and efficiency of growing microgreens in urban farming.*

Methods. *In order to make informed decisions in the climate control system, the article examines the Q-learning algorithm, which consists of such stages as determining different climatic states of the system; selecting the action to be performed based on the current state of the system and a utility estimate that is calculated based on the Bellman equation. A microclimate management model was developed and implemented, which uses the Q-learning algorithm to optimize climate parameters. The research methodology included simulation of various environmental conditions, model training based on collected data and experimental testing in real conditions of urban farming.*

Results. *Experimental simulations using the Python programming language with TensorFlow, PyTorch and scikit-learn libraries confirmed the effectiveness of applying the Q-learning algorithm in the climate control system to increase the productivity and efficiency of growing microgreens. To ensure that the system has reached the desired state, strategies such as monitoring the actual parameter values using IoT sensors of the climate control system, analyzing the obtained Q-table values, and setting learning stopping criteria are used. The results of the program are transmitted to the actuators via the Wi-Fi data network using the ESP8266 microcontroller, which is used as a Wi-Fi module for the Arduino microcontroller.*

Conclusions. *The use of a climate control system with the Q-learning algorithm in urban farming contributes to the achievement of greater productivity, efficiency and stability of plant cultivation, which is reflected in the improvement of the results of plant cultivation.*

Keywords: *distributed systems, IoT technologies, cloud computing, Q-learning, monitoring.*

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

УДК 519.816
DOI: https://doi.org/10.17721/AIT.2024.1.03

Євген ІВОХІН, д-р фіз.-мат. наук, проф.
ORCID ID: 0000-0002-5826-7408
e-mail: ivohin@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Костянтин ЮШТІН, канд. фіз.-мат. наук, докторант
ORCID ID: 0009-0001-9881-2343
e-mail: gkons@mail.univ.kiev.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

ВИКОРИСТАННЯ МУРАШИНОГО АЛГОРИТМУ ДЛЯ РОЗВ'ЯЗАННЯ НЕЧІТКОЇ ЗАДАЧІ КОМІВОЯЖЕРА

В с т у п . ***Сформульовано та наведено методику пошуку оптимальної тривалості маршруту для розв'язання задачі комівояжера у випадку визначення часу переміщення між містами у вигляді нечітких трапецієподібних чисел. Метою роботи є розроблення алгоритму на основі оптимізації колонії мурах і використання цього методу для розв'язування задач комівояжера з достатньо великою кількістю міст транспортної мережі.***

М е т о д и . ***Використано метод на основі алгоритму оптимізації мурашиної колонії.***

Р е з у л ь т а т и . ***Для досягнення поставленої мети запропоновано схему реалізації оптимізаційного алгоритму, що за умови невеликої кількості ітерацій дозволяє отримувати наближені до оптимальних розв'язків результати пошуку шляхів у нечіткій задачі комівояжера. Запропонований підхід може бути використаний для пошуку раціонального шляху в ситуаціях із неточно заданою тривалістю переміщень між містами. Показано, що вибір основних параметрів алгоритму оптимізації колонії мурах суттєво не впливає на якість отриманого наближеного розв'язку. Приклади використання алгоритму підтверджують конструктивність підходу для розв'язання задачі комівояжера у випадку нечітко заданої тривалості переміщень.***

В и с н о в к и . ***Запропоновано схему реалізації алгоритму оптимізації мурашиної колонії для пошуку найкращого шляху в задачі комівояжера зі змінною тривалістю переміщень між містами, розроблено комп'ютерну програму, яка дозволяє розв'язувати різні логістичні задачі, в основу яких покладено задачу комівояжера з нечітко визначеними параметрами руху у транспортній мережі.***

К л ю ч о в і с л о в а : ***нечітка задача комівояжера, оптимізаційний метод мурашиної колонії, трапецієподібні нечіткі числа, дефазифікація, оцінювання ефективності.***

Вступ

Один із перспективних напрямів науково-практичних досліджень соціальних та інформаційних процесів базується на використанні математичних методів, у яких закладено принципи реалізації природних механізмів для прийняття рішень. Ройовий інтелект – це відносно новий технологічний підхід, який формалізується на основі аналізу соціальної поведінки тварин і комах. Зокрема і спостереження за мурахами дозволили розробити ряд методів і прийомів, серед яких найбільш вивченим і найуспішнішим є метод оптимізації загального вигляду, відомий як оптимізація колонії мурах (Ant System Optimization, ASO). Імітація самоорганізації мурашиної колонії становить основу мурашиних алгоритмів оптимізації – нового перспективного методу природних обчислень. Іншими природними прототипами для оптимізаційних методів також можуть бути: поведінка бабок (алгоритм рою бабок, BFO), бджіл (алгоритм бджолиного рою, BA), термітів (алгоритм термітів), риб (алгоритм рибної зграї, FSO) та вовків (алгоритм вовчої зграї, WSA).

Під реалізацією ройового інтелекту розуміють спосіб розв'язування різноманітних оптимізаційних проблем за допомогою групи агентів, які взаємодіють між собою згідно з простими правилами, за якими функціонує складна поведінка всієї системи. Стосовно його використання для методики оптимізації найголовнішою перевагою є можливість знаходити глобальні оптимуми у задачах із великою кількістю параметрів та обмежень, а також гнучкість, масштабованість, можливість розподілених обчислень і захист від відмов. Системи на базі ройового інтелекту дозволяють оперативно знаходити ефективні розв'язки за умов динамічних змін параметрів, відмов окремих агентів, а також ці системи не потребують задання умов централізованого керування.

З іншого боку, серед недоліків ройового інтелекту варто зазначити складність налаштування параметрів, що збільшує ризик знаходження локальних оптимумів оптимальних роз'вязків, високі вимоги до обчислювальних ресурсів і необхідність експериментального оцінювання.

Розглянемо застосування ройового інтелекту для розв'язування задач оптимізації на прикладі використання ASO (Dorigo, Maniezzo, & Colorni, 1996). Мурашині алгоритми широко використовуються вченими із середини 90-х рр. Першу версію алгоритму запропонував Марко Доріго 1992 р. (Dorigo, Maniezzo, & Colorni, 1991). Нині вже отримано непогані результати мурашиної оптимізації для розв'язання таких складних комбінаторних завдань: задачі оптимізації маршрутів вантажівок, завдання розмальовки графа, квадратична задача про призначення, оптимізація мережних графіків, задачі календарного планування тощо (Bell, 2004; Zhao, Luo, & Zhang, 2010; Raspinelli, 2002). Особливо ефективними є мурашині алгоритми за online оптимізації процесів у розподілених нестаціонарних системах, наприклад, для розв'язання проблем розподілу трафіка у телекомунікаційних мережах (Schoonderwoerd, 1996).

Колонія мурах може розглядатися як багатоагентна система, в якій кожен агент (мураха) функціонує автономно за визначеними правилами. Поведінка кожного агента обумовлена простими випадковими правилами. Цей принцип збігається з поведінкою мурах у реальному світі, де вони працюють разом для будівництва гнізд, пошуку їжі та захисту колонії. У роботах (Bonavear, 1999; Bullnheimer, 1999) показано, що на базі примітивної поведінки окремих агентів, поведінка сукупної системи дозволяє отримати найкращі результати для різного класу задач.

Ідея алгоритму сформована на основі поведінки мурашиної колонії, яка знаходить шлях до їжі, близький до оптимального. Основу поведінки мурах складає самоорганізація – сукупність динамічних механізмів, за допомогою яких система досягає глобальної мети в результаті взаємодії елементів на низькому рівні. Принциповою особливістю такої низькорівневої взаємодії є використання елементами системи лише локальної інформації, без будь-яких правил

централізованого керування та звернення до даних, що описують глобальні параметри зовнішнього середовища у вигляді обмежень. Самоорганізація агентів є наслідком взаємодії таких чотирьох компонентів:

- випадковість;
- додатний зворотний зв'язок;
- від'ємний зворотний зв'язок;
- багатократність взаємодій.

Багатократність взаємодії реалізується у формі послідовного ітераційного пошуку маршруту одночасного декількома мурахами. Кожна мураха починає власний рух випадковим чином, коли покидає мурашник у пошуках їжі. Вважають, що кожен агент не рухається певним відомим шляхом або заздалегідь відомим напрямком. Ця експлоративна поведінка дозволяє мурахам досліджувати широку ділянку навколо мурашника. Ключовим аспектом поведінки мурах є здатність залишати на власному шляху хімічні сліди – феромони. Ці феромони слугують сигналами для інших мурах і вказують на те, що шлях вже був досліджений і використовується. Наприклад, у класичній задачі пошуку маршруту комівояжера на мережі, заданій у вигляді графа, додатний обернений зв'язок реалізується таким стохастичним правилом: "імовірність включення ребра графа в маршрут мурахи пропорційна кількості феромона на ній". Коли мураха знаходить їжу, вона повертається до мурашника і залишає феромоновий слід, який допомогає іншим мурахам знаходити шлях до їжі. Кількість феромона, який відкладає мураха на етапі маршруту, є обернено пропорційною до довжини відповідної ділянки. Чим коротший шлях у процесі пошуку знайшла мураха, тим більше феромона буде відкладено на відповідних етапах маршруту (ребрах графа).

Зауважимо, що використання лише додатного оберненого зв'язку веде до швидкої (передчасної) збіжності алгоритму, тобто до випадку, коли усі мурахи рухаються тим самим субоптимальним маршрутом. Для запобігання перенасиченню шляхів феромони із часом випаровуються, що реалізує від'ємний обернений зв'язок. Це дозволяє мурахам адаптуватись до змін у навколишньому середовищі (напр., поява нових джерел їжі або завад на шляхах). Випаровування феромонів забезпечує динамічне оновлення інформації та дозволяє уникнути вибору неоптимальних шляхів (уникнення локальних оптимумів). Час випаровування феромона не повинен бути надто великим для запобігання загрозі збіжності маршрутів руху всіх мурах до одного субоптимального розв'язку. З іншого боку, час випаровування не має бути й малим, щоб не призвести до некооперативної поведінки мурах через втрату пам'яті колонії.

Завдяки цим механізмам міжагентної взаємодії система саморганізується і дозволяє окремим мурахам використовувати оптимальні шляхи до джерел їжі з можливістю ефективного розв'язання задач розподілу та пошуку ресурсів без централізованого керування.

З математичного погляду модель ASO описується через такі базові компоненти, що пов'язані з поведінкою як окремих мурах, так і системи в цілому:

- механізми формування шляхів;
- розміщення феромонів для позначення пройдених шляхів;
- випаровування феромонів;
- правила вибору шляху мурахою.

Однією із задач, для якої можна запропонувати спосіб розв'язання на основі мурашиного алгоритму, є логістична задача комівояжера (Dantzig, 1954). ***Метою*** вказаного дослідження є адаптація алгоритму ASO для розв'язання задачі комівояжера (ЗК) з нечітко заданими параметрами переміщень на етапах транспортної мережі, досягнення якої пов'язано з проведенням ***завдань*** формалізації нечіткої задачі на основі використання нечітких трапецієподібних чисел, реалізації компонентів самоорганізаційної поведінки мурах для оптимізації маршруту комівояжера та проведення чисельних експериментів для визначення ефективності розробленого методу.

Постановка задачі. Стандартна постановка задачі комівояжера полягає у виборі найкоротшого за довжиною або часом замкненого шляху на мережі з n міст, що проходить через кожне місто рівно один раз. Кількість можливих варіантів дорівнює $(n-1)!$, а за умов симетричності етапів маршруту кількість унікальних маршрутів становить $(n-1)!/2$.

Знаходження оптимального шляху комівояжера, навіть у випадку відносно невеликої транспортної мережі у 60–70 міст, потребує надзвичайно великої кількості обчислювальних ресурсів для розрахунку, що призводить до необхідності використання наближених алгоритмів, таких як ASO.

У реальних задачах логістики поняття тривалості або вартості подорожі між окремими пунктами транспортної мережі не можуть бути фіксованими, вони визначається наближено, часто із впливом суб'єктивного фактора щодо оцінок часових термінів або вартості переміщення за ділянками маршруту. Це призводить до необхідності врахування невизначеності, її формалізації на основі різних методик. Одним із підходів, що використовуються у цьому випадку, є залучення нечітких чисел і реалізація засобів маніпуляції з ними.

Розглянемо постановку задачі комівояжера з нечітко заданою тривалістю переміщень на транспортній мережі. У такому випадку необхідно знайти циклічну перестановку номерів міст, які має відвідати комівояжер, відповідно до якої затрати часу будуть мінімальні з урахуванням обмеження щодо відвідування кожного з пунктів не більше одного разу. Математичне формулювання нечіткої задачі комівояжера можна записати так: потрібно мінімізувати відповідно до вказаного вище способу порівняння нечітких чисел цільову функцію

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{t}_{ij}x_{ij}\ , \tag{1}$$

де часові витрати на переміщення між пунктами задаються у вигляді матриці $\tilde{T}=\{\tilde{t}_{ij}\}$, $i,j=\overline{1,n}$, з елементами у вигляді нечітких чисел (Kumar, 2011), а можливі шляхи переміщень між містами визначаються матрицею X, за умови виконання обмежень:

$$\sum_{i=1}^{n}x_{ij}=1 \text{ для всіх } j=1,2,...,n, \qquad \sum_{j=1}^{n}x_{ij}=1 \text{ для всіх } i=1,2,...,n, \tag{2}$$

ISSN 2788-6603

$$i-j+nx_{ij}\le n-1,\ 1\le i\ne j\le n,$$

$x_{ij}=0$ або 1 для всіх $i,j=1,2,...,n$.

Для програмної реалізації у матриці $\tilde{T}$ діагональні елементи $\tilde{t}_{ii}$ необхідно задати великими додатними числами, щоб отримувати у розв'язку величини $x_{ii}=0$ для всіх $i=1,2,....,n$.

Нечіткі величини тривалості переміщення $\tilde{t}_{ii}$ між довільними містами $i,j=\overline{1,n}$, задаватимемо у вигляді трапецієподібних нечітких чисел.

Означення. Нечітким трапецієподібним числом $\tilde{A}$ (Kumar, 2011) називають упорядковану четвірку дійсних чисел (a_1,a_2,a_3,a_4), $a_1\le a_2\le a_3\le a_4$, для яких визначено функцію належності $\mu_{\tilde{A}}(x)$ вигляду

$$\mu_{\tilde{A}}(x)=\begin{cases}\dfrac{x-a_1}{a_2-a_1}, \text{якщо } a_1\le x\le a_2;\\ 1, \text{ якщо } a_2\le x\le a_3;\\ \dfrac{a_4-x}{a_4-a_3}, \text{ якщо } a_3\le x\le a_4.\end{cases} \tag{3}$$

Якщо до подання трапецієподібного нечіткого числа застосувати підхід на основі гауссівського розподілу з відповідними характеристиками, то в узагальненому випадку трапецієподібне нечітке число можна представити дещо в іншому вигляді як

$$\tilde{A}=(a_1,a_2,a_3,a_4)=([a_2,a_3],\alpha,\beta)=(m,w,\alpha,\beta), \tag{4}$$

де використовується середня точка $m=\dfrac{a_2+a_3}{2}$ та півширина плато $w=\dfrac{a_3-a_2}{2}$, а коефіцієнти $\alpha=a_2-a_1$ та $\beta=a_4-a_3$ визначають лівий і правий розподіл нечіткого числа $\tilde{A}=(a_1,a_2,a_3,a_4)$ відповідно.

Для оперування з нечіткими числами потрібно визначити операції, з огляду на наведений вище опис. За середню точку беруть звичайне середньоарифметичне значення границь плато, лівий і правий розподіли розглядають відповідно до правила ґратки, за яким для довільних дійсних чисел a,b покладемо $a\cup b=\max\{a,b\}$ та $a\cap b=\min\{a,b\}$.

Тоді для довільних трапецієподібних нечітких чисел $\tilde{A}=(m(\tilde{A}),w(\tilde{A}),\alpha_1,\beta_1)$ та $\tilde{B}=(m(\tilde{B}),w(\tilde{B}),\alpha_2,\beta_2)$ можна визначити операції додавання, віднімання, множення та ділення, які у загальному випадку позначимо символом $\circ$:

$$\begin{aligned}\tilde{A}\circ\tilde{B}&=(m(\tilde{A})\circ m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)=\\&=(m(\tilde{A})\circ m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\end{aligned} \tag{5}$$

Остаточно маємо

$$\begin{aligned}\tilde{A}+\tilde{B}&=(m(\tilde{A})+m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)=\\&=(m(\tilde{A})+m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\\\tilde{A}-\tilde{B}&=(m(\tilde{A})-m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)=\\&=(m(\tilde{A})-m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\\\tilde{A}\times\tilde{B}&=(m(\tilde{A})\times m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)=\\&=(m(\tilde{A})\times m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\\\tilde{A}\div\tilde{B}&=(m(\tilde{A})\div m(\tilde{B}),w(\tilde{A})\cup w(\tilde{B}),\alpha_1\cup\alpha_2,\beta_1\cup\beta_2)=\\&=(m(\tilde{A})\div m(\tilde{B}),\max(w(\tilde{A}),w(\tilde{B})),\max(\alpha_1,\alpha_2),\max(\beta_1,\beta_2)).\end{aligned}$$

Для проведення операцій порівняння та ранжування нечітких чисел використаємо спосіб на основі медіанного середнього значення. Іншими словами, якщо для кожного $\tilde{A}=(a_1,a_2,a_3,a_4)\in F(R)$ визначено функцію ранжування

$\Re: F(R) \to R$ із медіанним середнім значенням у вигляді $\Re(\tilde{A}) = \left[\left(\frac{a_2 + a_3}{2}\right) + \left(\frac{\beta - \alpha}{4}\right)\right]$, тоді для довільних двох трапецієподібних нечітких чисел $\tilde{A} = (a_1, a_2, a_3, a_4)$ та $\tilde{B} = (b_1, b_2, b_3, b_4)$ маємо такі можливі варіанти порівняння:

- $\tilde{A} \succ \tilde{B}$ тоді і лише тоді, якщо $\Re(\tilde{A}) > \Re(\tilde{B})$;
- $\tilde{A} \prec \tilde{B}$ тоді і лише тоді, якщо $\Re(\tilde{A}) < \Re(\tilde{B})$;
- $\tilde{A} \approx \tilde{B}$ тоді і лише тоді, якщо $\Re(\tilde{A}) = \Re(\tilde{B})$.

Процеси оброблення нечітких чисел передбачають етап дефазифікації – перетворення нечіткого результату до чіткого (числового) значення. Це важливий крок у методиці застосування нечіткого підходу, особливо в задачах нечіткого керування та нечіткої бізнес-логіки, де потрібно перетворити нечіткі розв'язки на конкретні події або числові значення. Існують різні методи дефазифікації, серед яких найпоширенішими є: метод центра тяжіння (Center of Gravity, CoG) або центроїду, метод середнього максимуму та метод максимуму. Для порівняння результатів дослідження використовуватимемо метод центра тяжіння. У цьому методі точка дефазифікації обчислюється як центр тяжіння нечіткої множини. Для випадку неперервного способу подання нечітких чисел формула розрахунку центра тяжіння має вигляд

$$CoG = \frac{\int_{a_1}^{a_4} x \cdot \mu(x)\, dx}{\int_{a_1}^{a_4} \mu(x)\, dx}, \tag{6}$$

а для дискретної нечіткої множини:

$$CoG = \frac{\sum_{i=1}^{n} \mu(x_i) \cdot x_i}{\sum_{i=1}^{n} \mu(x_i)}, \tag{7}$$

де x_i – точки, що визначають результат, а $\mu(x_i)$ – ступінь належності кожної точки нечіткій множині, $i = \overline{1,n}$.

Методи

Формування шляхів у моделі оптимізації колонії мурах описують за допомогою графа. Нехай $G = (V, E)$ являє собою граф, де V – це множина з m вершин, а E – множина ребер. Кожному ребру, що з'єднує вершини i та j графа G, $(i, j) \in E$, ставлять у відповідність два параметри: час переміщення T_{ij} по ребру (i, j) (зазвичай пропорційний довжині шляху D_{ij}) та інтенсивність феромона τ_{ij} на ребрі (i, j), $i, j = m$.

Збереження рівня феромонів є ключовим процесом для міжагентної взаємодії мурах між собою. Інтенсивність феромонів на ребрі τ_{ij} оновлюється на основі досвіду мурах, що пройшли за цим ребром. Для оновлення рівня феромона часто використовують формулу

$$\tau_{ij}(s+1) = (1-\rho) \cdot \tau_{ij}(s) + \Delta\tau_{ij}(s),$$

де ρ – коефіцієнт випаровування феромона, $0 < \rho < 1$, а $\Delta\tau_{ij}(s)$ – кількість феромона, що залишається мурахами на ребрі (i, j) на ітерації s, $s = 0, 1, 2, \ldots$.

Випаровування феромонів зменшує інтенсивність феромона на всіх ребрах, що дозволяє системі забувати попередні (можливо неоптимальні) шляхи й адаптуватись до змін. Це запобігає передчасній збіжності до локальних оптимумів.

Вибір кожного етапу шляху руху мурахи базується на випадкових правилах. Коли на ітерації s мураха k перебуває у вершині i, вона обирає наступну вершину j з певною ймовірністю $P_{ij}^k(s)$, яка залежить від інтенсивності феромона і видимості η_{ij}, яка є обернено пропорційною довжині шляху $\eta_{ij} = \frac{1}{D_{ij}}$:

$$\begin{cases} P_{ij}^k(s) = \dfrac{(\tau_{ij}(s))^{\alpha} (\eta_{ij})^{\beta}}{\sum_{l \in J_i^k} (\tau_{il}(s))^{\alpha} (\eta_{il})^{\beta}}, \text{ якщо } j \in J_i^k; \\ P_{ij}^k(s) = 0, \text{ якщо } j \notin J_i^k, \end{cases} \tag{8}$$

де α та β – регульовані параметри, що контролюють відносний вплив феромона та видимості, відповідно, а сума у знаменнику розраховується за всіма доступними ребрами (множина доступних вершин J_i^k з вершини i). Якщо α = 0, то найвірогіднішим буде перехід у найближчі міста. У класичній теорії оптимізації це відповідає так званому, жадібному

ISSN 2788-6603

алгоритму. Якщо β = 0, тоді працює лише феромонне підсилення, що призводить до швидкого завершення роботи алгоритму через збіжність маршрутів усіх мурах до одного субоптимального розв'язку.

Зазначимо, що сума всіх імовірностей переходу з вершини i за всіма можливими варіантами з множини J_i^k на ітерації s дорівнює 1:

$$\sum_{j \in J_i} P_{ij}^k(s) = 1. \tag{9}$$

Основна мета мурашиного алгоритму – мінімізація довжини шляху L, який є сумою довжин ребер на шляху мурахи:

$$L = \sum_{(i,j) \in P} D_{ij}, \tag{10}$$

де P позначає множину ребер, з яких складається шлях мурахи.

Розглянемо схему реалізації чотирьох основних компонентів самоорганізаційної поведінки мурах під час оптимізації маршруту комівояжера. Послідовно реалізуючи ітераційні кроки, що відтворюють процедуру знаходження маршруту кожною мурахою, отримуємо схему функціонування мурашиного алгоритму для розв'язання нечіткої задачі комівояжера. Перехід мурахи з міста i в місто j на ітерації s алгоритму залежить від трьох складових: табу-списку, видимості та віртуального сліду феромона. Табу-список A_i^k – це перелік міст, які вже відвідані мурахою k до вершини i і заходити в які ще раз заборонено. Цей перелік збільшується з проходженням за маршрутом та очищується на початку кожної ітерації алгоритму. Позначимо через J_i^k перелік міст, які ще потрібно відвідати мурасі k, що перебуває у місті i. Зрозуміло, що об'єднання переліків A_i^k та J_i^k дає множину всіх міст, указаних у задачі комівояжера.

На першій ітерації $s = 0$ кількість феромонів на кожному шляху $\tau_{ij}(0) = 0$.

Генерується набір мурах в кожній вершині графа.

В процесі ітерації кожна мураха діє окремо. В першому місті i табу-перелік A_i^k для мурахи k складається з міста, в якому перебуває мураха $-A_i^k = \{i\}$. Далі кожна мураха обирає з певною ймовірністю наступне місто для переміщення, з урахуванням заданої часової тривалості пересування до найближчих міст за формулою (1). Для цього використовуватимемо генератор випадкових чисел.

Після обрання наступного міста j, додаємо це місто до табу-переліку $A_j^k = \{i, j\}$. На наступному кроці обираємо наступне місто для переходу і так далі до останнього міста в маршруті. Якщо маршрут не можна замкнути, то мураха вважається недієвою до наступної ітерації.

Після завершення ітерації розраховуємо випаровування феромона для кожного можливого ребра. Далі для кожної мурахи з успішно завершеним маршрутом після виконання ітерації s розраховуємо тривалість маршруту і для кожного ребра, використаного в маршруті, додаємо феромон у кількості, яка є обернено пропорційною тривалості маршруту для кожної мурахи:

$$\Delta\tau_{ij}(s) = \frac{Q}{L_k},$$

де L_k – тривалість успішного маршруту кожної мурахи k, а ребро $(i, j) \in L_k$ належить до маршруту мурахи k; Q – регульований параметр, значення якого обирають одного порядку з довжиною оптимального маршруту (Bullnheimer, 1999).

Отже, за умови коректно обраних значень параметрів α та β з кожною ітерацією поступово отримаємо покращені результати.

Результати обчислень

Наведемо результати проведених чисельних експериментів. На попередньому етапі алгоритм ASO був адаптований до розв'язання задачі комівояжера з нечіткою тривалістю переміщень між містами. Для цього виконано фазифікацію задачі, наведеної в роботі [Ivohin, 2023] за допомогою трапецієподібних нечітких чисел (рис. 1).

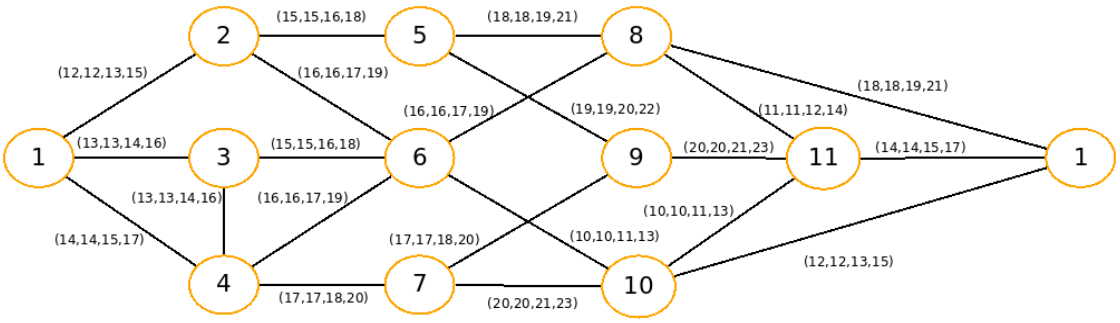


Рис. 1. Схема-приклад задачі комівояжера з нечіткою тривалістю

Ранжування маршрутів із нечіткою тривалістю виконували за допомогою методу обчислення COG у вигляді (6) для неперервного способу подання нечітких чисел.

Для даної задачі проведено чисельні розрахунки маршруту комівояжера методом повного перебирання та за допомогою алгоритму ASO. Застосування алгоритму ASO для задачі комівояжера у вказаній конфігурації характеризується високою швидкістю збіжності та забезпечує отримання найкращого результату за досить консервативних параметрів ASO (табл. 1).

Таблиця 1

Параметри алгоритму ASO

α	β	Q	V_{vap}	**Ітерація**
0,1	–1	20	0,05	4
0,1	–2	20	0,05	3
0,1	–4	20	0,05	2
0,15	–1	20	0,05	5
0,2	–1	20	0,05	5
0,25	–1	20	0,05	7
0,3	–1	20	0,05	8

У стовпці "Ітерація" вказано першу ітерацію, на якій отримано найкращий розв'язок із найбільшою ймовірністю.

Оптимальна тривалість маршруту в задачі комівояжера на основі повного перебирання визначається нечітким трапецієподібним числом (156, 156, 167, 189). Будемо вважати цей розв'язок точним. Під час застосування мурашиного алгоритму для розв'язання задачі отримано аналогічний розв'язок, причому результат досягається, зазвичай, вже на 3-й або 4-й ітерації, а його дефазифікована на основі методу CoG величина складає 167,92 одиниць.

Подальші експерименти з ASO проводили для оцінювання якості отриманого результату з урахуванням різної кількості міст транспортної мережі. У чисельних розрахунках генетичний алгоритм використовували для розв'язання ЗК у разі випадкового розміщення $n = 16, 17, 18$ міст на двовимірній площині 200 x 200, для якого величини переміщення між містами мережі визначались за середньою тривалістю нечіткого часу, яка вважалась пропорційною відстані між відповідними містами. Проведено три варіанти чисельних експериментів: кількість ітерацій алгоритму ASO дорівнює чотирикратній кількості міст (табл. 2), подвійній кількості міст (табл. 3), кількості міст (табл. 4). Кількість мурах дорівнює кількості міст у кожному випадку.

Для кожного експерименту проведено оцінювання впливу зміни параметрів β та α на остаточний результат. Порівняння найкращого результату, отриманого за допомогою алгоритму ASO за задану кількість ітерацій, з найкращим результатом, отриманим методом повного перебирання, наведено в табл. 2, 3 та 4. Значення якості порівняння результатів подано у відсотках, які визначаються величиною отриманого збільшення довжини шляху щодо результату, який отримано методом повного перебирання.

Таблиця 2

Параметри наближеного оптимального розв'язку (кількість ітерацій дорівнює чотирикратній кількості міст)

V_{vap}	α	β	Q	**Максимальне відносне відхилення від точного розв'язку для різної кількості міст, %**		
				16	**17**	**18**
0,05	0,15	–1	20	3,72	4,31	4, 0
0,05	0,15	–1,5	20	3,56	3,81	4,28
0,05	0,15	–2	20	3,67	3,73	4,19
0,05	0,15	–2,5	20	3,75	3,93	3,97
0,05	0,15	–3	20	3,68	4,18	4,27
0,05	0,15	–3,5	20	3,97	4,21	4,30
0,05	0,15	–4	20	3,88	4,11	4,33
0,05	0,1	–3	20	3,77	4,04	4,64
0,05	0,15	–3	20	3,68	3,78	3,87
0,05	0,2	–3	20	3,66	3,71	3,98
0,05	0,25	–3	20	4,02	4,10	4,19
0,05	0,3	–3	20	4,08	4,11	4,32

ISSN 2788-6603

Таблиця 3

Параметри наближеного оптимального розв'язку (кількість ітерацій дорівнює подвійній кількості міст)

V_{vap}	α	β	Q	Максимальне відносне відхилення від точного розв'язку для різної кількості міст, %		
				16	17	18
0,05	0,15	–1	20	4,04	4,50	4,74
0,05	0,15	–1,5	20	3,94	4,48	4,78
0,05	0,15	–2	20	3,98	4,23	4,81
0,05	0,15	–2,5	20	3,99	4,17	4,26
0,05	0,15	–3	20	3,97	3,99	4,56
0,05	0,15	–3,5	20	4,10	4,26	4,38
0,05	0,15	–4	20	3,86	4,36	4,55
0,05	0,1	–3	20	3,88	4,11	4,58
0,05	0,15	–3	20	3,87	3,91	4,56
0,05	0,2	–3	20	4,01	4,40	4,47
0,05	0,25	–3	20	4,12	4,28	4,83
0,05	0,3	–3	20	4,34	4,44	4,72

Таблиця 4

Параметри наближеного оптимального розв'язку (кількість ітерацій дорівнює кількості міст)

V_{vap}	α	β	Q	Максимальне відносне відхилення від точного розв'язку для різної кількості міст, %		
				16	17	18
0,05	0,15	–1	20	4,75	4,92	4,96
0,05	0,15	–1,5	20	4,61	5,06	5,20
0,05	0,15	–2	20	4,42	4,45	4,69
0,05	0,15	–2,5	20	4,35	4,69	5,11
0,05	0,15	–3	20	4,54	4,57	4,75
0,05	0,15	–3,5	20	4,38	4,39	4,71
0,05	0,15	–4	20	4,11	5,12	5,15
0,05	0,1	–3	20	4,37	4,50	5,13
0,05	0,15	–3	20	4,34	4,47	4,75
0,05	0,2	–3	20	4,31	4,77	4,83
0,05	0,25	–3	20	4,45	4,70	4,83
0,05	0,3	–3	20	4,47	4,96	5,05

Як видно з наведених результатів за умови здійснення кількості ітерацій, що дорівнює кількості міст, алгоритм дозволяє отримати результати, що лежать у межах 5 % від найкращого. Збільшення кількості ітерацій веде до покращення результатів. Варіація параметра β в межах від –4 до –1 та параметра α в межах від 0,1 до 0,3 якісно впливають на результати, які у цьому випадку розміщені в межах 0,5 % від найкращого результату розв'язування задачі комівояжера, який отримано методом повного перебирання.

Дискусія і висновки

В роботі розглянуто та проаналізовано застосування алгоритму оптимізація колонії мурах для розв'язання задачі комівояжера з нечітко заданою тривалістю переміщень на транспортній мережі. Запропоновано схему реалізації алгоритму, що за умови невеликої кількості ітерацій дозволяє отримувати наближені до оптимальних розв'язків результати. Запропонований підхід може бути використаний для пошуку найкращого шляху в ситуаціях зі змінною тривалістю переміщень між містами. Показано, що вибір основних параметрів алгоритму оптимізації колонії мурах суттєво не впливає на якість отриманого наближеного розв'язку. Наведено приклади використання алгоритму, що підтверджують конструктивність підходу для знаходження розв'язків задачі комівояжера у випадку задання нечіткої тривалості переміщень.

Внесок авторів: Євген Івохін – розроблення методики та методології дослідження, написання висновків; Костянтин Юштін – огляд літературних джерел, адаптація алгоритму, проведення емпіричних досліджень та опис результатів.

Список використаних джерел

Bell, J., & McMullen, P. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1), 41–48.

Bonavear, E., & Dorigo, M. (1999). *Swarm Intelligence: from Natural to Artificial Systems*. Oxford University Press.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A New Rank-Based Version of the Ant System: A Computational Study. *Central European Journal for Operations Research and Economics*, 1(7), 25–38.

Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1954). Solution of a Large-scale Traveling Salesman Problem. *Operations Research,* 2.

Dorigo, M., Maniezzo, V., & Colorni, A. (1991). *Positive feedback as a search strategy* Dipartimento di Elettronica, Politecnico di Milano,Tech. Rep. 91-016.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). *The Ant System: Optimization by a colony of cooperating agents.* IEEE Transactions on Systems, Man, and Cybernetics. Part B. *Cybernetics*, 26(1), 29–41.

Ivohin, E. V., Gavrylenko, V. V., & Ivohina, K. E. (2023). On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method. *Radioelectronics, Computer Science, Control,* 3, 141–147.

Kumar, A., & Gupta, A. (2011). Methods for solving fuzzy assignment problems and fuzzy travelling salesman problems with different membership functions. *Fuzzy Information and Engineering*, 3(1), 3–21.

Raspinelli, J. M., Lopes, H. S., & Freitas, A. A. (2002). Data Mining with an Ant Colony Optimization Algorithm. *IEEE Trans. On Evolutionary Computation. Special issue on Ant Colony Algorithms*, 6(4), , 321–332.

Schoonderwoerd, R., Holland, O., Bruten, J., & Rothkranz, L. (1996). *Ant-based load balancing in telecommunication networks*. Adaptive Behavior, 5(2).

Zhao, D., Luo, L., & Zhang, K. (2010). An improved ant colony optimization for the communication network routing problem. *Mathematical and Computer Modelling*, 52(11–12), 1976–1981.

References

Bell, J., & McMullen, P. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1), 41–48.

Bonavear, E., & Dorigo, M. (1999). *Swarm Intelligence: from Natural to Artificial Systems*. Oxford University Press.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). A New Rank-Based Version of the Ant System: A Computational Study. *Central European Journal for Operations Research and Economics*, 1(7), 25–38.

Dantzig, G. B., Fulkerson, D. R., & Johnson, S. M. (1954). Solution of a Large-scale Traveling Salesman Problem. *Operations Research,* 2.

Dorigo, M., Maniezzo, V., & Colorni, A. (1991). *Positive feedback as a search strategy* Dipartimento di Elettronica, Politecnico di Milano,Tech. Rep. 91-016.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). *The Ant System: Optimization by a colony of cooperating agents.* IEEE Transactions on Systems, Man, and Cybernetics. Part B. *Cybernetics*, 26(1), 29–41.

Ivohin, E. V., Gavrylenko, V. V., & Ivohina, K. E. (2023). On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method. *Radioelectronics, Computer Science, Control,* 3, 141–147.

Kumar, A., & Gupta, A. (2011). Methods for solving fuzzy assignment problems and fuzzy travelling salesman problems with different membership functions. *Fuzzy Information and Engineering*, 3(1), 3–21.

Raspinelli, J. M., Lopes, H. S., & Freitas, A. A. (2002). Data Mining with an Ant Colony Optimization Algorithm. *IEEE Trans. On Evolutionary Computation. Special issue on Ant Colony Algorithms*, 6(4), 321–332.

Schoonderwoerd, R., Holland, O., Bruten, J., & Rothkranz, L. (1996). *Ant-based load balancing in telecommunication networks*. Adaptive Behavior, 5(2).

Zhao, D., Luo, L., & Zhang, K. (2010). An improved ant colony optimization for the communication network routing problem. *Mathematical and Computer Modelling*, 52(11–12), 1976–1981.

Отримано редакцією журналу / Received: 21.08.24
Прорецензовано / Revised: 19.09.24
Схвалено до друку / Accepted: 25.10.24

Eugene IVOHIN, DSc (Phys. & Math.), Prof.
ORCID ID: 0000-0002-5826-7408
e-mail: ivohin@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Kostyantyn YUSHTIN, Doctoral Student
ORCID ID: 0009-0001-9881-2343
e-mail: gkons@mail.univ.kiev.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

USE OF ANT COLONY OPTIMIZATION ALGORITHM FOR SOLVING FUZZY PROBLEM OF TRAVELING SALESMAN

Background. ***The method of finding the optimal route length for the traveling salesman problem in the case of determining the time of movement between cities in the form of fuzzy trapezoidal numbers is formulated and given. The purpose of the work is to develop an algorithm based on the optimization of an ant colony and use this method to solve problems of a traveling salesman with a sufficiently large number of cities in the transport network.***

Methods. ***The method based on the ant colony optimization algorithm was used.***

Results. ***In order to achieve the goal, a scheme for the implementation of the optimization algorithm is proposed, which, under the condition of a small number of iterations, allows obtaining results close to optimal solutions in the vague problem of the traveling salesman. The proposed approach can be used to find a rational path in situations with an imprecisely specified duration of movements between cities. It is shown that the selection of the main parameters of the ant colony optimization algorithm does not significantly affect the quality of the obtained approximate solution. Examples of the use of the algorithm confirm the constructiveness of the approach to solving the traveling salesman problem in the case of a vaguely specified duration of movements.***

Conclusions. ***A scheme for the implementation of the ant colony optimization algorithm for finding the best path in the problem of a traveling salesman with variable duration of movements between cities is proposed, a computer program has been developed that allows solving various logistics problems, which are based on the problem of a traveling salesman with vaguely defined movement parameters in the transport network.***

Keywords: ***fuzzy traveling salesman problem, ant colony optimization method, trapezoidal fuzzy numbers, defuzzification, performance evaluation.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

УДК 519.816
DOI: https://doi.org/10.17721/AIT.2024.1.04

Олексій КУЗНЄЦОВ, асп.
ORCID ID: 0000-0002-3537-9976
e-mail: oleksiy1908@gmail.com
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

Геннадій КИСЕЛЬОВ, канд. техн. наук, ст. наук. співроб.
ORCID ID: 0000-0003-2682-3593
e-mail: g.kyselov@gmail.com
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

ЗАСТОСУВАННЯ ТА АНАЛІЗ ФОРМАЛЬНИХ МЕТОДІВ ОЦІНЮВАННЯ РЕЛЕВАНТНОСТІ АВТОМАТИЧНО СТВОРЕНИХ РЕФЕРАТІВ ІНФОРМАЦІЙНИХ ТЕКСТІВ

Вступ. ***Розглянуто існуючі підходи до оцінювання якості автоматично створених рефератів інформаційних текстів. Дано огляд методів автоматичного реферування, включаючи класичні підходи та сучасні моделі на основі штучного інтелекту. Огляд містить екстрактивні методи реферування, такі як TF-IDF та PageRank, а також графові методи, зокрема TextRank. Особливу увагу приділено абстрактним підходам, що включають моделі Generative Pretrained Transformer (GPT) і Bidirectional and Auto-Regressive Transformers (BART). Оцінювання якості генерованих рефератів виконують за допомогою кількісних метрик оцінювання релевантності рефератів, зокрема і ROUGE та BLEU.***

Методи. ***Проаналізовано кілька підходів до автоматичного реферування текстів. Класичні екстрактивні методи, зокрема і TF-IDF, обчислюють важливість термів на основі частоти їхнього вживання в документі та в колекції документів. PageRank і TextRank використовують графові моделі для визначення значущості речень на основі зв'язків між ними. Абстрактні методи, такі як GPT і BART, генерують нові речення, що стисло передають зміст оригінального тексту. Оцінювання ефективності кожного підходу здійснюється метриками ROUGE і BLEU, які вимірюють збіг між автоматично згенерованими рефератами й еталонними текстами. Особливу увагу приділено аналізу їхньої точності, гнучкості, вимогам до ресурсів і простоті реалізації.***

Результати. ***Результати дослідження свідчать, що метрики ROUGE показують хорошу точність у вимірюванні збігів n-грам (послідовностей з n слів), тоді як BLEU ефективна у завданнях машинного перекладу, але може не враховувати деякі синтаксичні особливості тексту. Оцінювання методів автоматичного реферування за допомогою цих метрик показала, що екстрактивні методи реферування, такі як TF-IDF, є ефективними для оброблення простих текстів, але можуть втратити важливий контекст у складних текстах. PageRank і TextRank дозволяють враховувати зв'язки між реченнями, проте можуть давати менш релевантні результати для текстів із слабко вираженими структурними зв'язками. Абстрактні моделі GPT і BART забезпечують гнучкіший підхід до реферування, створюючи нові речення, що краще передають зміст, однак потребують значних обчислювальних ресурсів і складні у впровадженні.***

Висновки. ***Поєднання класичних і сучасних методів автоматичного реферування текстів дозволяє досягти вищої якості результатів. Важливо враховувати специфіку тексту та вимоги до кінцевого результату, адаптуючи обрані підходи та метрики відповідно до завдання.***

Ключові слова: ***автоматичне реферування, екстрактивні методи, абстрактні методи, GPT, BART, ROUGE, BLEU, TextRank, PageRank, TF-IDF.***

Вступ

У сучасному світі інформація є одним із найцінніших ресурсів. Щодня створюють величезні обсяги нових даних – від наукових статей до новинних повідомлень, технічної документації та багато іншого. У такому інформаційному потоці швидкий доступ до релевантної інформації стає надзвичайно важливим. Тут на допомогу приходять реферати, які дозволяють отримати суть документа без необхідності читати його повністю.

Реферати виконують важливу роль, особливо у наукових дослідженнях, де вони допомагають дослідникам швидко зрозуміти, чи є певний документ корисним для їхньої роботи. Однак створення рефератів вручну вимагає багато часу і зусиль, що не завжди можливо через великі обсяги даних. Ця проблема стає особливо гострою в умовах швидкого зростання інформаційних потоків, коли необхідність автоматизованого реферування стає очевидною. Для оцінювання результатів реферування використовують відповідні метрики, які дозволяють визначити, наскільки точно і повно реферат передає зміст оригінального документа.

Класичні методи автоматичного реферування – екстрактивні методи, де важливі речення просто витягуються з тексту, й абстрактні методи, що вимагають створення нових речень на основі розуміння змісту, мають свої переваги та недоліки. З появою штучного інтелекту з'явилися нові підходи до реферування, що базуються на глибинному навчанні й нейронних мережах, які обіцяють значне покращення якості автоматично створених рефератів. Однак, попри значні досягнення, задача автоматичного реферування текстів не може вважатись розв'язаною.

Постановка задачі. Основна мета цієї статті полягає в аналізі існуючих метрик оцінювання релевантності автоматично створених рефератів відносно оригінальних інформаційних текстів. На основі експериментів із використання цих метрик сформульовано рекомендації щодо застосування методів автоматичного реферування інформаційних текстів. Для досягнення цієї мети необхідно розв'язати такі задачі.

1. Проаналізувати наявні екстрактивні методи реферування текстів, такі як TF-IDF, PageRank і TextRank, й оцінити їхні переваги та недоліки в різних контекстах застосування.
2. Проаналізувати існуючі абстрактні методи автоматичного реферування текстів, такі як GPT та BART.
3. Дослідити можливості й обмеження використовуваних кількісних метрик оцінювання релевантності рефератів, зокрема ROUGE та BLEU.
4. Провести порівняльний аналіз різних методів реферування з використанням кількісних метрик оцінювання релевантності рефератів
5. Розробити рекомендації щодо вибору оптимальних методів реферування текстів різних типів, враховуючи особливості задачі та вимоги до точності, контекстуальності й обчислювальних витрат.

Ця робота націлена на пошук балансу між точністю, ресурсною ефективністю та якістю автоматично згенерованих рефератів, що дозволить покращити ефективність оброблення текстової інформації у різних сферах застосування.

Методи

В роботі використано екстрактивні й абстрактні методи автоматичного реферування текстів. Розглянемо їх по черзі.

Екстрактивні методи

Автоматичне реферування текстів є важливим інструментом оброблення великих обсягів інформації для забезпечення швидкого доступу до ключових даних. В основу класичних підходів до автоматичного реферування покладено методи, які дозволяють автоматично витягувати найважливіші речення з тексту, зберігаючи при цьому оригінальний стиль і формулювання автора. Ці методи відомі як екстрактивні та використовують різні техніки й алгоритми для визначення значущості та релевантності речень у контексті всього документа (Kuznietsov, & Kyselov, 2024).

Класичні екстрактивні методи включають TF-IDF (Term Frequency-Inverse Document Frequency), алгоритми ранжування, такі як PageRank, графові методи, наприклад, TextRank, та різноманітні статистичні підходи. Кожен із цих методів має свої унікальні характеристики, що дозволяють ефективно виокремлювати ключову інформацію з тексту.

Розуміння принципів роботи вказаних методів, їхніх переваг та обмежень є важливим для подальшого вдосконалення автоматичних систем реферування. Розглянемо основні класичні підходи до автоматичного реферування текстів, їхні особливості та вплив на якість створених рефератів. Це дозволить оцінити ефективність таких методів і визначити напрямки для подальших досліджень і розвитку технологій автоматичного реферування текстів.

- *Екстрактивний метод* ***TF-IDF***

TF-IDF (*частота терміна – обернена частота документа*) – це статистичний показник, що використовується для визначення значущості терміна в межах одного документа з певного набору документів.

TF (*частота терміна*) визначає відношення кількості появи слова до загальної кількості слів у документі (формула 1). Це дозволяє оцінити важливість терміна *t* в конкретному документі *d* (Кузнєцов, Кисельов, 2022).

$$tf(t,d) = n_t / \sum_k n_k, \quad (1)$$

де n_t – кількість входжень терму t в документ; $\sum_k n_k$ – загальна кількість слів у документі.

IDF (*обернена частота документа*) відображає частоту, з якою певне слово зустрічається у всіх документах колекції (формула 2). IDF знижує вагу поширених слів, надаючи перевагу унікальним термінам. Для кожного унікального слова в межах набору документів існує одне значення IDF. Терміни з високою частотою в конкретному документі та низькою частотою в інших документах отримують вищий бал у TF-IDF.

$$idf(\mathrm{t}, D) = log\left(|D| \Big/ \left|\left\{d_i \in |D| \mathrm{t} \in d_i\right\}\right|\right), \quad (2)$$

де $|D|$ – кількість документів у колекції; $\left|\left\{d_i \in |D| \mathrm{t} \in d_i\right\}\right|$ – кількість документів із колекції D, в яких зустрічається t (коли $n_t \neq 0$).

Відношення правдоподібності (log-likelihood ratio) є логарифмом відношення ймовірності спостереження слова з однаковою ймовірністю у корпусі вхідних документів і відповідних їм резюме до ймовірності появи слова з різними ймовірностями в цих корпусах (Lin, & Hovy, 2000).

Переваги методу TF-IDF

1. *Простота реалізації:* TF-IDF легко зрозуміти і впровадити. Його математична основа є інтуїтивно зрозумілою і базується на простих статистичних принципах.

Приклад. У будь-якій мові програмування, як-от Python, реалізація TF-IDF може бути здійснена за допомогою кількох рядків коду з використанням бібліотек, напр., scikit-learn.

2. *Ефективність для великих колекцій тексту:* TF-IDF добре масштабується для оброблення великих наборів даних і колекцій документів. Він дозволяє швидко й ефективно аналізувати і витягувати ключові слова з великої кількості текстів.

Приклад. Пошукові системи, такі як Google, використовують модифіковані версії TF-IDF для швидкого індексування і ранжування вебсторінок, дозволяючи користувачам отримувати релевантні результати пошуку за лічені секунди.

3. *Врахування значущості слів:* TF-IDF дозволяє оцінити важливість термінів у контексті документа, виділяючи ключові слова, які мають значення для конкретного документа.

Приклад. У статті про технології слова "штучний інтелект" і "машинне навчання" отримають високі значення TF-IDF, оскільки вони є ключовими термінами, що описують тему статті.

4. *Зменшення ваги поширених слів:* TF-IDF автоматично знижує вагу поширених слів, таких як сполучники і прийменники, які не несуть значної інформації.

Приклад. У новинній статті слово "повідомляють" матиме низьку вагу TF-IDF, оскільки воно часто зустрічається у багатьох новинах, тоді як специфічні терміни, такі як "економічна криза", отримають вищу вагу.

Недоліки методу TF-IDF

1. *Відсутність семантичного контексту:* TF-IDF не враховує значення слів у контексті, що може призводити до неправильної інтерпретації багатозначних слів.

Приклад. У реченнях "банк річки" і "банк фінансової установи" TF-IDF не розрізняє різні значення слова "банк", оскільки метод базується лише на частоті слів без урахування їхнього контексту.

2. *Чутливість до частоти:* якщо слово часто зустрічається у багатьох документах, його вага може бути знижена навіть у контексті документа, де воно є ключовим.

Приклад. Якщо слово "технології" часто зустрічається в наборі документів, його вага може бути знижена навіть у документах, де воно є основною темою.

ISSN 2788-6603

3. *Ігнорування порядку слів:* TF-IDF не враховує порядок слів у тексті, що може призводити до втрати сенсу під час оброблення текстів.

Приклад. Фрази "машинне навчання" і "навчання машинне" матимуть однакове значення TF-IDF, хоча порядок слів має значення для розуміння сенсу.

4. *Неадаптивність до нових слів:* TF-IDF може неадекватно оцінювати нові або рідкісні слова, оскільки метод потребує достатньої кількості документів для оцінювання частоти.

Приклад. Якщо нове слово, як-от "криптовалюта", з'являється вперше в новому документі, TF-IDF може не надати йому високої ваги, оскільки метод потребує достатньої кількості документів для адекватного оцінювання частоти.

- *Екстрактивний метод у вигляді алгоритму ранжування PageRank*

PageRank – це алгоритм, розроблений Ларрі Пейджем і Сергієм Бріном 1996 р., який використовують для оцінювання важливості вебсторінок у пошукових системах. Основна ідея PageRank полягає в тому, що важливість сторінки визначається не лише кількістю посилань на неї, але й важливістю сторінок, які на неї посилаються (Mihalcea, & Tarau, 2004).

Формула для розрахунку PageRank сторінки P виглядає так:

$$PR(P) = \frac{1-d}{N} + d\sum_{i=1}^{k}\frac{PR(P_i)}{L(P_i)}, \quad (3)$$

де $PR(P)$ – значення PageRank сторінки P; d – коефіцієнт затухання (зазвичай встановлюється на рівні 0,85); N – загальна кількість сторінок; P_i – сторінка, що посилається на сторінку P; $L(P_i)$ – кількість посилань, які виходять зі сторінки P_i.

Коефіцієнт затухання d відображає ймовірність того, що користувач продовжить переходити за посиланнями, а не завершить перегляд. Зазвичай d встановлюється на рівні 0,85, що означає, що користувач з імовірністю 85 % перейде за посиланням і з імовірністю 15 % почне перегляд нової сторінки.

Основні принципи PageRank

1. Графова модель:
   - Інтернет розглядається як граф, де вузли – це вебсторінки, а ребра – це гіперпосилання між сторінками.
2. Вага посилань:
   - Кожне посилання з однієї сторінки на іншу розглядається як "голос" за цю сторінку.
   - Важливіші сторінки отримують більше голосів, і, відповідно, їх PageRank зростає.
3. Розподіл ваги:
   - Кожна сторінка передає свою вагу (PageRank) сторінкам, на які вона посилається.
   - Вага розподіляється рівномірно між усіма вихідними посиланнями.
4. Ітеративний розрахунок:
   - PageRank обчислюється ітеративно, поки значення не закінчать збігання (перестануть значно змінюватися між ітераціями).

Переваги PageRank

1. *Об'єктивність оцінювання важливості:* PageRank оцінює важливість сторінки на основі її зв'язків з іншими сторінками. Це дозволяє алгоритму визначати важливість сторінок об'єктивно, без урахування їхнього вмісту.

Приклад. Уявімо два блоги: один має багато посилань від авторитетних новинних сайтів, інший – від особистих блогів. PageRank надасть перевагу першому блогу через високий авторитет його посилань, навіть якщо їхня кількість менша.

2. *Зважування якості посилань:* алгоритм враховує не лише кількість посилань, але й їхню якість. Посилання з авторитетних сторінок мають більшу вагу, ніж посилання з менш важливих сторінок.

Приклад. Сторінка, яка отримала посилання від урядового сайту, отримає більший приріст PageRank, ніж сторінка, яка отримала посилання від маловідомого блога.

3. *Зниження впливу спаму:* PageRank стійкий до маніпуляцій через спам-сайти, оскільки посилання з низькоякісних сторінок мають менший вплив.

Приклад. Якщо хтось створює багато фальшивих сайтів, що посилаються на основний сайт, PageRank основного сайту не зросте значно, тому що ці нові сайти мають низький PageRank.

4. *Гнучкість застосування:* алгоритм PageRank можна застосовувати не лише до вебсторінок, але й до інших структур, таких як наукові статті, соціальні мережі тощо.

Приклад. PageRank може використовуватися для оцінювання впливу наукових робіт на основі цитувань від інших наукових статей.

Недоліки PageRank

1. *Часові витрати на обчислення:* обчислення PageRank вимагає багато ітерацій для досягнення збіжності, що може бути часовитратним і ресурсомістким.

Приклад. Пошукова система, що індексує мільярди сторінок, потребує значного часу й обчислювальних ресурсів для регулярного оновлення значень PageRank.

2. *Проблеми з новими сторінками:* нові сторінки спочатку мають низький PageRank, навіть якщо вони містять якісний контент, через відсутність вхідних посилань.

Приклад. Новий блог з високоякісними статтями може довго залишатися непоміченим, поки він не отримає достатню кількість посилань від авторитетних джерел.

3. *Вразливість до посилальних ферм:* хоча PageRank стійкий до багатьох видів маніпуляцій, він може бути вразливий до посилальних ферм, де сайти взаємно посилаються один на одного для штучного підвищення їхнього PageRank.

Приклад. Група вебмайстрів може створити мережу сайтів, що взаємно посилаються один на одного, тим самим штучно підвищуючи їхній PageRank.

4. *Недостатня увага до контенту:* алгоритм зосереджується на зв'язках між сторінками, але не враховує якість їхнього контенту.

Приклад. Сторінка з великою кількістю вхідних посилань може мати високий PageRank, навіть якщо її контент застарілий або неякісний.

- *Екстрактивний метод на основі графів відомий як TextRank*

Методи на основі графів є потужним інструментом для аналізу тексту, структурування інформації та автоматичного реферування. Вони базуються на ідеї представлення тексту у вигляді графа, де вузли можуть представляти слова, речення або документи, а ребра – відношення між ними, такі як схожість, зв'язність чи послідовність.

TextRank – це один із популярних алгоритмів для автоматичного реферування текстів, який заснований на ідеях алгоритму PageRank, але адаптований для обробки тексту. Цей метод, як і PageRank, базується на ідеї графового представлення даних, де вузли представляють об'єкти (наприклад, слова або речення), а ребра – зв'язки між ними (Mihalcea, & Tarau, 2004).

Основні концепції TextRank

1. Графове представлення тексту:
   - Вузли: у методі TextRank текст представляється у вигляді графа, де вузли можуть бути або словами, або реченнями, залежно від задачі.
   - Ребра: ребра між вузлами відображають певний тип зв'язку між цими елементами тексту. Наприклад, для графа слів це може бути суміжність у тексті, а для графа речень – схожість у змісті.
2. Ранжування вузлів:
   - Подібно до PageRank, TextRank використовує ітеративний процес для ранжування вузлів на основі кількості та ваги посилань, що вказують на них. Чим більше важливих вузлів посилаються на даний вузол, тим більший ранг він отримує.
3. Збіжність ітерацій:
   - Процес ранжування триває, поки значення рангу не стабілізується (збіжність). На практиці, це означає, що після кількох ітерацій алгоритм припиняє роботу, коли зміни в ранжуванні вузлів стають мінімальними.

Переваги TextRank

1. *Мовна незалежність:* TextRank можна застосовувати до текстів різними мовами без необхідності модифікації алгоритму. Це можливо тому, що TextRank працює зі структурою тексту, а не з його змістом на семантичному рівні.

Приклад. Ви можете використовувати TextRank для автоматичного реферування статей на англійській, українській, китайській чи будь-якій іншій мові з мінімальними змінами. Алгоритм просто будує граф слів або речень і аналізує їхні зв'язки, незалежно від мови.

2. *Автоматичне визначення значущості:* TextRank автоматично визначає важливі елементи тексту (речення або слова) на основі їхньої зв'язності з іншими елементами, що дозволяє отримати об'єктивні результати без необхідності навчання на великих наборах даних.

Приклад. Якщо у вас є наукова стаття, TextRank може виділити основні тези і висновки, які часто згадуються у тексті та мають сильний зв'язок з іншими важливими реченнями.

3. *Простота реалізації:* TextRank є відносно простим для реалізації, оскільки він не вимагає попереднього навчання моделей або наявності великих обсягів анотованих даних.

Приклад. Упровадити TextRank у додаток для оброблення тексту можна за допомогою готових бібліотек або, якщо написати кілька рядків коду. Це особливо корисно для невеликих проєктів або стартапів, які не мають ресурсів для складних моделей машинного навчання.

4. *Застосування до різних задач:* TextRank можна використовувати як для реферування текстів, так і для екстракції ключових слів. Це універсальний інструмент для аналізу тексту.

Приклад. Для блога можна автоматично згенерувати короткий підсумок статті або виділити ключові слова для SEO-оптимізації, використовуючи той самий алгоритм.

Недоліки TextRank

1. *Чутливість до вибору метрик і налаштувань:* якість роботи TextRank залежить від вибору метрик для вимірювання схожості між елементами тексту й інших налаштувань алгоритму. Неправильний вибір може призвести до неточних результатів.

Приклад. Якщо використовувати невідповідну метрику схожості між реченнями, TextRank може виявити не ті речення як найважливіші, що призведе до некоректного реферату. Наприклад, речення з однаковими стоп-словами можуть бути помилково визнані схожими.

2. *Ігнорування глибокого контексту:* TextRank не враховує глибокі семантичні зв'язки між словами або реченнями, що може обмежити його ефективність для складних текстів.

Приклад. У художньому тексті, де важлива контекстуальна інформація і метафори, TextRank може не виділити ключові речення правильно, оскільки він не розуміє семантичний зміст тексту, а лише його поверхневу структуру.

3. *Обмеження у довгих текстах:* TextRank може погано працювати з дуже довгими текстами, де кількість зв'язків між елементами стає занадто великою, що може призвести до зниження ефективності та збільшення обчислювальної складності.

Приклад. Якщо спробувати застосувати TextRank до великого роману, алгоритм може неефективно визначити важливі частини тексту через велику кількість взаємозв'язків між реченнями, що робить реферат непослідовним або неповним.

4. *Залежність від суміжності:* TextRank часто визначає важливість елементів на основі їхнього суміжного положення в тексті. Це може бути проблематично, коли важливі ідеї розподілені у тексті нерівномірно.

Приклад. У статті, де ключова ідея розкидана по різних частинах тексту, TextRank може не з'єднати ці фрагменти в один важливий вузол, ігноруючи в такий спосіб основну думку.

Абстрактні методи

Абстрактні методи є складною та потужною технологією у сфері автоматичного реферування текстів. Вони відрізняються від екстрактивних методів тим, що не просто вибирають найважливіші фрази або речення з оригінального тексту, а створюють нові, узагальнювальні резюме, які можуть містити нові слова та структури речень (Moratanch, & Chitrakala, 2016). Ключові особливості абстрактних методів виглядають так.

1. *Генерація нового тексту:* абстрактні методи здатні створювати новий текст, який не просто повторює слова та фрази з вихідного документа, а передає його зміст у більш стислій і зручній для сприйняття формі.

ISSN 2788-6603

Це означає, що модель може перефразовувати, узагальнювати і навіть включати додатковий контекст для створення більш чіткого і зв'язного резюме. Наприклад, якщо в оригінальному тексті детально описується певна подія, абстрактний метод може згенерувати коротке резюме, яке передає сутність цієї події, без повторення всіх деталей.

Приклад. Якщо оригінальний текст містить кілька абзаців, що описують наукове дослідження і його результати, абстрактне резюме може виглядати так: "Дослідження показало, що новий метод лікування значно покращує виживаність пацієнтів із серцевою недостатністю".

2. *Глибоке розуміння контексту:* щоб створити змістовне резюме, абстрактні методи повинні розуміти зміст тексту на глибокому рівні, включаючи семантичні зв'язки між різними частинами тексту.

Ці методи часто використовують складні моделі машинного навчання, такі як трансформери або рекурентні нейронні мережі, які навчаються на величезних обсягах даних. Моделі здатні вивчати не тільки прямі зв'язки між словами, але й контекст, у якому вони вживаються, що дозволяє їм створювати більш узгоджені і зв'язні резюме.

Уявіть, що модель обробляє текст, де описують кілька взаємопов'язаних подій. Абстрактний підхід може зв'язати ці події в одне узагальнене резюме, що передає основну ідею, наприклад: "Ланцюг подій привів до значного зростання економіки країни".

3. *Використання нейронних мереж:* абстрактні методи часто базуються на нейронних мережах, зокрема і на моделях типу трансформерів, які є основою для багатьох сучасних систем оброблення природної мови (NLP).

Такі моделі, як GPT (Generative Pretrained Transformer) або BART (Bidirectional and Auto-Regressive Transformers), спочатку навчаються на великих обсягах тексту, щоб зрозуміти структуру і зміст природної мови. Після цього вони можуть генерувати новий текст, що відображає головні ідеї оригінального документа, але у стислій формі.

Використовуючи GPT, модель може проаналізувати статтю про нову технологію і згенерувати резюме: "Нова технологія штучного інтелекту покращує точність діагностики медичних зображень".

4. *Постпроцесинг для покращення якості:* після того, як текст генерується, він проходить етап постпроцесингу, щоб переконатися, що резюме відповідає граматичним, стилістичним і змістовим стандартам.

На цьому етапі можуть бути використані додаткові модулі або алгоритми, які перевіряють текст на узгодженість, логічність, коректність стилю, усувають можливі помилки або некоректності.

Якщо модель згенерувала резюме з неузгодженостями або орфографічними помилками, ці проблеми будуть виправлені на етапі постпроцесингу, щоб отримати кінцевий якісний генеративний (Generative) результат.

5. *Навчання на великих обсягах даних:* щоб досягти високої якості результатів, абстрактні методи потребують значних обсягів текстових даних для навчання.

Чим більше текстів використовують для навчання моделі, тим краще вона розуміє різні стилі письма, контексти та теми. Це дозволяє абстрактним методам створювати резюме, які є точнішими та змістовнішими.

Якщо модель навчалася на мільйонах статей і книг, вона буде ефективнішою у створенні резюме для нових текстів у різних галузях, таких як медицина, наука, література тощо.

Абстрактні методи автоматичного реферування тексту є потужним інструментом для створення стиснених резюме, які зберігають основний зміст оригіналу, але формулюються новими словами та фразами. Ці методи використовують складні нейронні мережі для аналізу і розуміння тексту, що дозволяє їм генерувати нові тексти, які є більш змістовними і компактними порівняно з вихідними. Однак такі методи потребують великих обчислювальних ресурсів і великих обсягів даних для навчання, що робить їхню реалізацію складнішою, ніж екстрактивні методи.

Трансформери для моделювання природної мови

Generative Pretrained Transformer

Generative Pretrained Transformer (генеративний трансформер із попереднім навчанням) – це потужна архітектура для моделювання природної мови, створена на основі трансформерів. Ця модель була розроблена компанією OpenAI і стала основою багатьох сучасних систем оброблення природної мови (NLP). Розглянемо детальніше ключові аспекти GPT (OpenAI, 2022):

1. *Архітектура трансформера:* модель GPT базується на архітектурі трансформера, яку запропоновано в статті "Attention is All You Need" (2017) (Vaswani et al., 2017). Трансформери використовують механізм уваги (attention) для оброблення послідовностей даних, що дозволяє їм ефективно враховувати контекст на всіх рівнях тексту.

Увага дозволяє моделі зосередитися на важливих частинах вхідних даних, що особливо важливо для завдань, які вимагають розуміння контексту. Наприклад, якщо модель аналізує речення, увага дозволяє їй зосередитися на тих словах, які є найбільш релевантними для поточного слова або фрази.

2. *Переднавчання (Pretraining):* модель GPT спочатку навчається на величезних обсягах текстових даних, використовуючи неконтрольоване навчання. Це означає, що модель просто намагається передбачити наступне слово в тексті, враховуючи попередні слова. Цей процес дозволяє моделі вивчити структуру мови, граматичні правила, семантику та загальні закономірності.

Переднавчання дозволяє моделі отримати базові знання про мову, що робить її здатною виконувати різні завдання оброблення природної мови після додаткового навчання.

3. *Генерація тексту:* після переднавчання GPT може використовуватися для генерування тексту. Коли модель отримує початковий текст (контекст), вона може передбачати та додавати нові слова, фрази або навіть абзаци, продовжуючи текст у стилі та темі вихідного матеріалу.

GPT здатна генерувати довгі та зв'язні тексти на основі лише кількох початкових слів або речень. Це робить її дуже корисною для завдань, які вимагають креативного підходу або автоматизації написання тексту, наприклад, для створення новин, оповідань, резюме або відповідей на запитання.

4. *Навчання з перенесенням (Transfer Learning):* після переднавчання модель може бути додатково налаштована (fine-tuned) на конкретні завдання, використовуючи менший набір даних. Наприклад, якщо потрібно створити модель для генерації юридичних текстів, модель GPT може бути навчена на корпусі юридичних документів.

Навчання з перенесенням дозволяє ефективно адаптувати модель до нових завдань, зберігаючи знання, отримані під час переднавчання. Це значно підвищує продуктивність моделі та робить її гнучкою для використання в різних галузях (Hosna et al., 2022).

5. *Шкала та потужність:* з кожною новою версією GPT (GPT-2, GPT-3, GPT-4) модель стає більшою, з більшою кількістю параметрів. Наприклад, GPT-3 має 175 мільярдів параметрів, що робить її однією з найбільших і найпотужніших моделей у сфері оброблення природної мови.

Велика кількість параметрів дозволяє моделі краще узагальнювати знання та точно виконувати завдання, навіть якщо вони суттєво відрізняються від того, на чому модель навчалася.

6. *Застосування:*

- *Чат-боти та віртуальні асистенти:* GPT використовують для створення чат-ботів, які можуть вести природні діалоги з користувачами, відповідаючи на запитання та допомагаючи з різними завданнями.
- *Автоматичне написання текстів:* модель може генерувати статті, резюме, електронні листи, технічну документацію та багато іншого.
- *Творче письмо:* GPT здатна створювати поеми, сценарії та інші творчі тексти на основі певної тематики або стилю.

Переваги GPT

1. *Здатність генерувати високоякісний текст:* GPT може генерувати зв'язний, змістовний і граматично правильний текст.

Приклад. GPT може створювати статті, що виглядають як написані людиною. Наприклад, на основі запиту "Опишіть значення штучного інтелекту в медицині" GPT може згенерувати детальну статтю з описом впливу ШІ на діагностику та лікування пацієнтів. Це робить GPT корисним для автоматизації написання текстів, таких як новини, блоги або резюме.

2. *Універсальність і гнучкість:* GPT може бути використаний у багатьох завданнях, як-от переклад, створення резюме, чат-боти, відповіді на запитання і багато іншого.

Приклад. У сценарії використання в чат-ботах GPT може підтримувати природну розмову з користувачами, відповідати на їхні запитання та надавати інформацію на основі введених даних. Одна модель може бути налаштована на різні завдання, що робить її надзвичайно гнучкою.

3. *Широке переднавчання:* GPT навчається на величезних обсягах текстових даних, що дозволяє їй отримати знання з багатьох галузей.

Приклад. GPT може генерувати текст на різноманітні теми – від науки до мистецтва. Наприклад, GPT може написати есе на тему філософії або технічний звіт про новітні досягнення у сфері квантових обчислень. Це робить GPT здатною адаптуватися до різних запитів і надавати релевантну інформацію в різних контекстах.

4. *Можливість налаштування на специфічні завдання:* GPT можна додатково налаштовувати на конкретні завдання або теми за допомогою навчання на менших, спеціалізованих наборах даних.

Приклад. Якщо потрібно, щоб GPT створювала юридичні документи, її можна налаштувати на наборі юридичних текстів, щоб модель стала експертом у цій галузі. Це дозволяє отримати більш точні та спеціалізовані результати в конкретних сферах.

Недоліки GPT

1. *Ризик генерації некоректної інформації:* GPT може генерувати текст, який виглядає правдоподібно, але містить фактичні помилки або некоректну інформацію.

Приклад. На запит "Скільки планет у Сонячній системі?" GPT може помилково згенерувати відповідь, що включає планети, яких не існує. Це робить GPT ненадійним у випадках, коли потрібна точна і перевірена інформація, особливо в наукових або технічних контекстах.

2. *Відсутність справжнього розуміння контексту:* GPT не має справжнього розуміння контексту або намірів, що може призводити до випадкових або недоречних відповідей.

Приклад. Якщо GPT запитати про етичні аспекти технологій, то система може згенерувати поверхневий або навіть безглуздий текст, оскільки модель не розуміє етичні концепції на глибинному рівні. Це може обмежувати здатність GPT генерувати текст, який вимагає глибокого розуміння складних концепцій або контекстів.

3. *Проблеми з упередженнями:* GPT може відображати упередження, які містяться в навчальних даних, оскільки вона навчалася на текстах, написаних людьми з різними поглядами.

Приклад. Якщо GPT навчена на текстах з інтернету, вона може відтворювати упереджені думки або стереотипи. Наприклад, GPT може генерувати тексти, що містять гендерні або расові стереотипи. Це створює ризик поширення дезінформації або закріплення упереджень, що є небажаним в етичному контексті використання технологій.

4. *Велика потреба в обчислювальних ресурсах:* GPT потребує значних обчислювальних ресурсів для навчання і використання, особливо коли йдеться про великі моделі, такі як GPT-3.

Приклад. Щоб розгорнути GPT-3 для генерації тексту в реальному часі, може знадобитися потужний сервер або хмарні обчислювальні ресурси. Це може бути дорогим і недоступним для деяких користувачів або організацій, що обмежує можливості широкого впровадження цієї технології.

Generative Pretrained Transformer має значні переваги, такі як здатність генерувати високоякісний текст, універсальність у різних завданнях, широке переднавчання і можливість налаштування. Водночас, модель має і недоліки, серед яких ризик генерації некоректної інформації, відсутність справжнього розуміння контексту, можливість упереджень і висока потреба в обчислювальних ресурсах. Ці плюси та мінуси роблять GPT потужним інструментом, але вимагають обережного й усвідомленого підходу до її використання.

Bidirectional and Auto-Regressive Transformers

Bidirectional and Auto-Regressive Transformers (BART) або Gemini – це нейромережна модель, розроблена компанією Facebook AI (тепер Meta AI), яка поєднує два підходи до трансформерів: двонаправлений (bidirectional) та авторегресивний (auto-regressive). BART створено як універсальну модель для оброблення природної мови (NLP), що може виконувати різні завдання, такі як генерація тексту, перефразування, автоматичне реферування тощо (Lewis et al., 2020).

Основні концепції BART

1. *Двонаправленість (Bidirectional):* двонаправлені трансформери, такі як BERT (Bidirectional Encoder Representations from Transformers), читають текст повністю, аналізуючи контекст як зліва направо, так і справа наліво. Це дозволяє моделі краще розуміти значення слів у контексті (Devlin et al., 2019).

Приклад. У реченні "Кішка сидить на підвіконні і дивиться у вікно", двонаправлена модель аналізує всі слова одночасно, щоб зрозуміти, що "кішка" – це суб'єкт дії, а слова "сидить" та "дивиться" описують її дії.

ISSN 2788-6603

2. *Авторегресивність (Auto-Regressive):* авторегресивні моделі, такі як GPT, генерують текст послідовно, прогнозуючи наступне слово на основі попередніх. Це означає, що на кожному кроці модель бере до уваги тільки ті слова, які вже були згенеровані.

Приклад. Якщо GPT генерує речення "Погода сьогодні чудова", то після слова "погода" модель передбачає, яке слово може бути наступним, на основі попередніх слів.

3. *Комбінація двох підходів у BART:* BART поєднує двонаправлений підхід, як у BERT, для кодування тексту, і авторегресивний підхід, як у GPT, для декодування тексту. Це робить модель універсальною та здатною до виконання широкого спектра завдань оброблення природної мови.

Приклад. Для задачі автоматичного реферування BART може використовувати двонаправлений підхід, щоб повністю зрозуміти вихідний текст, і авторегресивний підхід, щоб згенерувати короткий зміст тексту, використовуючи цю інформацію.

BART має енкодер-декодерну архітектуру, де:

- енкодер працює в двонаправленому режимі, аналізуючи контекст і зліва направо, і справа наліво.
- декодер працює в авторегресивному режимі, генеруючи текст послідовно, слово за словом.

Це дозволяє BART ефективно працювати як для завдань розуміння тексту (напр., класифікація тексту), так і для завдань генерації тексту (напр., машинний переклад або автоматичне реферування).

Переваги BART

1. *Універсальність:* BART є дуже гнучкою моделлю, здатною виконувати широкий спектр задач оброблення природної мови – автоматичне реферування, текстове перефразування, машинний переклад і заповнення пропусків у тексті.

Приклад. Якщо вам потрібно перефразувати текст, ви можете використовувати BART, і він дасть альтернативний, але збережений за змістом варіант. Наприклад, для фрази "Штучний інтелект змінює медицину" BART може запропонувати варіант "Інтелектуальні системи революціонізують охорону здоров'я".

2. *Висока точність:* завдяки двонаправленому енкодеру, який аналізує текст у повному контексті, і авторегресивному декодеру, який генерує текст послідовно, BART досягає високої точності в багатьох задачах.

Приклад. У задачі автоматичного реферування BART може виділити найбільш значущі фрагменти тексту для створення зведеного викладу. Наприклад, із довгої статті про зміну клімату модель може створити стислий, але інформативний реферат.

3. *Ефективність у розв'язанні різних задач:* оскільки BART поєднує двонаправлені й авторегресивні підходи, він ефективний у різних задачах генерації і розуміння тексту.

Приклад. BART може бути використаний як для задачі класифікації тексту (де важливо розуміти контекст), так і для генерації тексту (напр., написання короткого опису до заданого контенту).

4. *Можливість fine-tuning:* BART можна адаптувати до конкретних завдань або доменів додатковим навчанням на специфічних наборах даних, що робить його універсальнішим.

Приклад. Якщо вам потрібна модель для автоматичного створення підсумків юридичних документів, ви можете налаштувати BART на спеціалізованому наборі юридичних текстів, щоб досягти високої точності в цьому завданні.

Недоліки BART

1. *Великі обчислювальні ресурси:* як і інші великі трансформерні моделі, BART вимагає значних обчислювальних ресурсів для навчання і використання, що може бути проблематично для невеликих компаній або дослідницьких груп.

Приклад. Навчання BART на великому корпусі текстів може зайняти багато часу та потребує потужного обладнання, такого як графічні процесори (GPU) або тензорні процесори (TPU).

2. *Можливість генерації некоректної інформації:* як і інші моделі генерації тексту, BART може створювати змістовні, але некоректні або непослідовні фрагменти тексту, особливо якщо дані навчання були неякісними або незбалансованими.

Приклад. Якщо модель неправильно інтерпретує контекст або отримує суперечливу інформацію, вона може згенерувати текст, який не відповідає реальності. Наприклад, у задачі створення новинного заголовка BART може створити сенсаційний, але неправдивий заголовок.

3. *Ризик перенавчання на специфічних даних:* якщо BART надмірно налаштовувати на специфічний набір даних, модель може втратити здатність генералізувати, тобто погано працювати на даних, які відрізняються від тих, на яких вона навчалася.

Приклад. Якщо BART налаштований тільки на тексти наукової літератури, він може погано справлятися з генерацією текстів для популярних медіа або реклами.

4. *Складність налаштування й інтерпретації:* висока складність моделі ускладнює її налаштування та розуміння, що може створювати труднощі для дослідників та інженерів за адаптації BART для специфічних задач.

Приклад. Для правильного налаштування BART під задачу автоматичного перекладу необхідні спеціалізовані знання в галузі оброблення природної мови, що може бути достатньо проблематичним для команд без відповідного досвіду.

Метрики для оцінювання релевантності автоматично генерованих рефератів

Кількісні метрики оцінювання релевантності рефератів є важливим інструментом для об'єктивного вимірювання якості автоматично генерованих рефератів порівняно з еталонними, створеними вручну. Такі метрики дозволяють оцінювати, наскільки точно та повно автоматичний реферат передає зміст оригінального тексту. Розглянемо основні кількісні метрики детальніше.

- *Метрики типу Recall-Oriented Understudy for Gisting Evaluation*

Recall-Oriented Understudy for Gisting Evaluation **(ROUGE)** – це набір метрик, розроблених для оцінювання якості автоматичних рефератів та інших текстів, що генеруються алгоритмами. Цю метрику широко використовують в обробленні природної мови (NLP), оскільки вона дозволяє порівнювати автоматично створені тексти з еталонними рефератами, написаними людьми.

Основна ідея ROUGE полягає в тому, щоб виміряти кількість спільних елементів (зазвичай це слова або фрази) між автоматичним і еталонним рефератом. Метрика визначає, наскільки добре автоматично створений текст відображає зміст еталонного тексту (Lin, 2004).

Види ROUGE

Існує кілька різновидів ROUGE, кожен з яких призначений для оцінювання різних аспектів збігу між текстами:

ROUGE-N

- ROUGE-N вимірює кількість збігів n-грам між автоматичним і еталонним рефератами. N-грам – це послідовність з N елементів (напр., слів), які розглядають як одне ціле.
- ROUGE-1: вимірює збіг уніграм, тобто окремих слів. Це найпростіший і найчастіше використовуваний варіант ROUGE.
- ROUGE-2: вимірює збіг біграм, тобто послідовностей із двох слів.
- ROUGE-3 і більше: можливі також триграми та інші n-грами, але вони використовуються рідше.

Формула:

$$ROUGE\text{-}N = \frac{\sum_{gram_n \in Reference} Count\_match(gram_N)}{\sum_{gram_n \in Reference} Count(gram_N)}, \quad (4)$$

де *Count_match* – кількість збігів *n*-грам в автоматичному рефераті, *Count* – загальна кількість *n*-грам в еталонному рефераті.

Приклад. Якщо еталонний реферат містить фразу "Зміна клімату викликає глобальні проблеми", а автоматичний реферат містить "Кліматичні зміни викликають глобальні проблеми", то для ROUGE-1 збіги включатимуть слова "клімат", "глобальні", "проблеми".

ROUGE-L

ROUGE-L вимірює довжину найдовшої загальної підпослідовності (LCS, Longest Common Subsequence) між автоматичним і еталонним текстами. Цей підхід враховує і слова, і порядок їхнього розташування в тексті.

Формула:

$$ROUGE\text{-}L = \frac{LCS}{Довжина\ еталонного\ тексту}, \quad (5)$$

де *LCS* – довжина найдовшої підпослідовності, яка зустрічається і в автоматичному, і в еталонному рефераті.

Приклад. Якщо еталонний реферат містить фразу "Танення льодовиків призводить до підвищення рівня моря", а автоматичний реферат містить "Підвищення рівня моря викликано таненням льодовиків", то LCS включатиме "Танення льодовиків... рівня моря", що показує збіг послідовності слів.

ROUGE-S (ROUGE-Skip)

ROUGE-S вимірює збіг так званих "сплітних" біграм. Цей підхід дозволяє враховувати збіги, коли слова в біграмі можуть бути розділені іншими словами.

Формула:

$$ROUGE\text{-}N = \frac{\sum_{skip_2 \in Reference} Count\_match(skip_2)}{\sum_{skip_2 \in Reference} Count(skip_2)}, \quad (6)$$

де *skip_2* – це біграма зі словом або словами між ними, які дозволено пропускати.

Приклад. Якщо еталонний реферат містить фразу "Глобальне потепління швидко прогресує", а автоматичний реферат містить "Швидке прогресування глобального потепління", ROUGE-S врахує збіги фраз типу "глобальне... потепління".

ROUGE-W (Weighted ROUGE)

ROUGE-W є розширенням ROUGE-L, яке враховує вагу збігу залежно від довжини підпослідовності. Цей підхід надає більшу вагу довшим спільним підпослідовностям, ніж коротким збігам.

Формула: залежить від обраної ваги для різних підпослідовностей, що дозволяє точніше налаштовувати метрику під конкретні задачі.

Переваги ROUGE

1. *Простота використання*

ROUGE легко розрахувати й інтерпретувати. Він оцінює, наскільки добре автоматичний реферат збігається з еталонним, на основі простих підрахунків збігів слів або фраз.

Приклад. Якщо ми маємо автоматичний реферат "Сонце сходить на сході", а еталонний – "Схід сонця на сході", то ROUGE-1 покаже високий рівень збігу, оскільки більшість слів повторюються, навіть якщо порядок дещо інший.

2. *Широке використання та підтримка*

ROUGE став стандартом де-факто для оцінювання якості автоматичних рефератів та інших текстових задач. Він підтримується багатьма інструментами та програмними бібліотеками, що робить його доступним для широкого кола дослідників і розробників.

Приклад. Використання ROUGE дозволяє легко порівнювати результати різних досліджень і алгоритмів, оскільки метрика широко прийнята в науковому співтоваристві. Наприклад, у порівнянні двох моделей реферування можна використати ROUGE для демонстрації того, яка модель краще генерує тексти.

3. *Гнучкість*

ROUGE може бути налаштований для різних типів тексту і завдань. Наприклад, можна використовувати ROUGE-1 для загального оцінювання збігу слів, ROUGE-2 для оцінювання послідовностей слів і ROUGE-L для оцінювання збігу підпослідовностей.

Приклад. Якщо потрібно оцінити реферати, які складаються з коротких ключових фраз, краще використати ROUGE-1 і ROUGE-2. А для довших текстів, де важливо врахувати послідовність і зв'язність, можна застосувати ROUGE-L.

4. *Незалежність від мови*

ROUGE можна застосовувати для текстів будь-якою мовою, що робить його універсальним інструментом для оцінювання якості текстів.

Приклад. Незалежно від того, чи працюєте ви з англійським, українським або китайським текстом, ROUGE можна використовувати без змін, що робить його зручним для міжнародних досліджень.

ISSN 2788-6603

Недоліки ROUGE

1. *Чутливість до лексичних відмінностей*

ROUGE орієнтований на точні збіги слів або фраз. Він не враховує синоніми або перефразування, що може призводити до заниження оцінювання для текстів, які передають однаковий зміст, але використовують різні слова.

Приклад. Автоматичний реферат "Сонце піднімається на сході", а еталонний реферат "Схід сонця на сході". Хоча зміст однаковий, ROUGE може дати низьку оцінку через відмінності у використаних словах, таких як "піднімається" і "схід".

2. *Неврахування семантики*

ROUGE оцінює лише поверхневі збіги, не враховуючи глибший зміст тексту. Це означає, що навіть якщо два тексти мають однакові ключові слова, але передають різні ідеї, ROUGE може показати високу оцінку.

Приклад. Два реферати можуть містити ті самі слова, але в різному контексті. Наприклад, у першому рефераті йдеться про "економічне зростання в Китаї", а в другому – про "економічне зростання у світі", хоча збіги слів можуть бути високими, зміст текстів різний, але ROUGE цього не врахує.

3. *Ігнорування синтаксису та контексту*

ROUGE не враховує синтаксичні зв'язки або контекстуальні взаємодії між словами, що може бути критичним для оцінкювання якості тексту.

Приклад. У фразі "Кіт побачив собаку" і "Собака побачив кота" порядок слів змінює значення, але ROUGE оцінить їх однаково, оскільки збіги слів однакові, ігноруючи зміну контексту.

4. *Чутливість до довжини тексту*

ROUGE може бути упередженим щодо довших текстів, оскільки більші тексти мають більше шансів на збіг з еталонними текстами просто через велику кількість слів.

Приклад. Довший реферат, навіть якщо він містить багато зайвих слів, може отримати вищу оцінку ROUGE, ніж короткий і точний реферат, оскільки в довшому тексті більше можливостей для збігу з еталонним текстом.

- *Метрики типу Bilingual Evaluation Understudy*

Bilingual Evaluation Understudy (BLEU) – це одна з найпопулярніших і найвідоміших метрик для автоматичного оцінювання якості машинного перекладу. Вона використовується для кількісного вимірювання того, наскільки близький згенерований переклад (кандидат) до одного або кількох еталонних перекладів, створених людиною (Papineni et al., 2002). Наведемо детальніший розгляд основних принципів і компонентів BLEU.

Основні принципи BLEU

1. *Оцінювання на основі n-грам:* BLEU оцінює схожість між кандидатським перекладом і еталонним за допомогою порівняння *n*-грам (послідовностей з *n* слів). *N*-грамна точність означає, що для певного *n* ми підраховуємо, скільки *n*-грам із кандидатського перекладу наявні в еталонному перекладі. Для BLEU зазвичай використовують *n*-грами від 1 до 4.

2. *Використання модифікованої точності:* BLEU використовує модифіковану *n*-грамну точність (modified precision), що означає, що для кожної з *n*-грам підраховується максимальна кількість її повторень в еталонному перекладі, щоб уникнути переоцінювання повторень у кандидатському перекладі. Це допомагає запобігти маніпуляціям із результатами шляхом багаторазового повторення певних слів або фраз у кандидатському перекладі.

3. *Штраф за кратність (Brevity Penalty):* BLEU включає штраф за кратність (brevity penalty), щоб враховувати різницю в довжині між кандидатським і еталонним перекладами. Штраф застосовують, коли кандидатський переклад коротший за еталонний, що запобігає генерації занадто коротких перекладів, які можуть мати високий збіг у *n*-грамі, але втрачати важливі аспекти змісту.

Основні компоненти BLEU

n-грамна точність (*n*-gram precision):

- 1-грамна точність (unigram precision) оцінює збіги окремих слів між кандидатським і еталонним перекладом;
- 2-грамна точність (bigram precision) оцінює збіги пар слів;
- 3-грамна точність (trigram precision) оцінює збіги трійок слів;
- 4-грамна точність (4-gram precision) оцінює збіги четвірок слів.

Загальний BLEU-бал розраховують як геометричне середнє від точності для всіх *n*-грам, що дозволяє оцінити відповідність перекладу як на рівні окремих слів, так і на рівні коротких фраз.

1. *Модифікована точність (Modified Precision):* щоб уникнути надмірної оцінки кандидатського перекладу за рахунок повторення тих самих слів, BLEU використовує модифіковану точність. Для кожного *n*-грам розраховують, скільки разів слово зустрічається в еталонному перекладі, і цю кількість збігів використовують для обчислення точності. Наприклад, якщо в кандидатському перекладі слово "кіт" зустрічається тричі, а в еталонному лише один раз, то для розрахунку модифікованої точності враховують лише один збіг.

2. *Штраф за кратність (Brevity Penalty):* Brevity Penalty використовують для запобігання генерації надто коротких перекладів, які можуть досягати високих показників точності за рахунок скорочення довжини (Callison-Burch, Osborne, Koehn, & 2006).

Формула для розрахунку штрафу така:

$$BP = \begin{cases} 1, \text{ якщо } c > r, \\ e^{\left(1-\frac{r}{c}\right)}, \text{ якщо } c \le r, \end{cases} \tag{7}$$

де c – довжина кандидатського перекладу, а r – довжина еталонного перекладу. Якщо кандидат довший або дорівнює еталону, штраф не застосовують.

3. *Геометричне середнє точності (Geometric Mean of Precision):* остаточний BLEU-бал розраховують як геометричне середнє від точності для кожної з *n*-грам, помножене на штраф за кратність. Це середнє забезпечує пропорційність між точністю для різних *n*-грам, дозволяючи оцінювати як збіг окремих слів, так і збіг коротких фраз.

Переваги BLEU

1. *Автоматизованість:* BLEU дозволяє автоматично оцінювати якість перекладу без необхідності залучати людських експертів.

Приклад. Припустимо, ви маєте тисячу перекладених речень і бажаєте швидко оцінити їхню якість. Використовуючи BLEU, можна швидко порівняти кандидатські переклади з еталонними, отримавши оцінку точності без витрат на ручну перевірку.

2. *Масштабованість:* BLEU добре працює з великими обсягами текстів, дозволяючи легко обробляти й оцінювати масштабні корпуси перекладів.

Приклад. В контексті онлайн-сервісу машинного перекладу, де щодня обробляють мільйони текстів, BLEU дозволяє постійно моніторити якість перекладу без значних затрат на ресурси.

3. *Універсальність:* BLEU можна використовувати для оцінювання результатів перекладів різними мовами та для різних типів текстів, від технічної документації до художньої літератури.

Приклад. Незалежно від того, чи перекладає система новини з англійської на французьку, чи наукові статті з китайської на іспанську, BLEU можна застосовувати для оцінювання якості перекладу.

4. *Урахування збігів n-грам:* BLEU враховує не тільки збіги окремих слів (1-грам), але й коротких фраз (2-грам, 3-грам тощо), що дозволяє оцінити не лише правильність окремих слів, а й збереження контексту.

Приклад. Якщо кандидатський переклад "The cat on mat" збігається з еталонним "The cat is on the mat" у 3-грамі ("The cat on"), BLEU покаже, що кандидат майже правильно передав основну ідею, хоч і пропустив деякі деталі.

Недоліки BLEU

1. *Ігнорування контексту:* BLEU оцінює тільки поверхневі збіги *n*-грам, ігноруючи глибокий зміст, семантику і контекст перекладу.

Приклад. Якщо переклад "He gave her cat food" замінити на "He gave her food for the cat", BLEU може знизити оцінку через відсутність точних збігів, хоча обидва варіанти мають однаковий зміст.

2. *Чутливість до варіативності:* BLEU погано враховує різні можливі варіанти правильного перекладу. Якщо еталонний переклад використовує одні слова, а кандидат – інші синонімічні, BLEU може дати низьку оцінку.

Приклад. Еталонний переклад: "The quick brown fox jumps over the lazy dog". Кандидат: "The fast brown fox leaps over the lazy dog". Незважаючи на те, що обидва переклади мають однаковий зміст, BLEU може знизити оцінку через відсутність збігів для слів "quick" і "jumps".

3. *Штраф за кратність (Brevity Penalty):* штраф за кратність може несправедливо знижувати оцінку, якщо кандидатський переклад коротший за еталонний, навіть якщо коротший переклад є точним і достатнім.

Приклад. Еталонний переклад: "The cat is on the mat". Кандидат: "The cat on mat". Незважаючи на те, що цей переклад є зрозумілим, BLEU може знизити оцінку через відсутність деяких слів і штраф за коротку довжину.

4. *Не враховує граматичні та синтаксичні помилки:* BLEU не враховує якість граматики та синтаксису в перекладі, тому текст із правильною послідовністю *n*-грам, але з поганою граматикою може отримати високу оцінку.

Приклад. Кандидат: "The cat on the mat is sitting". Якщо це порівнювати з еталоном "The cat is sitting on the mat", BLEU може показати високу оцінку, попри зміни порядку слів, що можуть порушувати граматику.

5. *Проблеми з довжиною n-грам:* використання великих *n*-грам може знижувати оцінки через те, що дуже рідко зустрічаються довгі послідовності, навіть якщо загальний зміст передано правильно.

Приклад. Якщо кандидатський переклад "The quick fox" збігається з еталонним "The quick brown fox", BLEU знизить оцінку через відсутність 3-грамного збігу "quick brown fox", навіть якщо втрачене слово не є критичним.

Результати використання метрик оцінювання релевантності текстів для розглянутих моделей

- *Приклад застосування моделі TF-IDF*

Розглянемо набір з трьох документів:

- Документ 1: "Штучний інтелект змінює світ".
- Документ 2: "Штучний інтелект у медицині і технологіях".
- Документ 3: "Технології і медицина розвиваються завдяки штучному інтелекту".

Розрахунок TF (частота терміна):

- В документі 1 слово "штучний" з'являється 1 раз, "інтелект" 1 раз, "змінює" 1 раз, "світ" 1 раз. Загальна кількість слів у документі 4.
- В документі 2 слово "штучний" з'являється 1 раз, "інтелект" 1 раз, "медицина" 1 раз, "технології" 1 раз. Загальна кількість слів у документі 5.
- В документі 3 слово "технології" з'являється 1 раз, "медицина" 1 раз, "розвиваються" 1 раз, "штучному" 1 раз, "інтелекту" 1 раз. Загальна кількість слів у документі 7.

Розрахунок IDF (зворотна частота документа):

- Для слова "штучний": з'являється у 2 документах з 3, IDF = log(3/2) = 0,176.
- Для слова "інтелект": з'являється у 2 документах з 3, IDF = log(3/2) = 0,176.
- Для слова "медицина": з'являється у 2 документах з 3, IDF = log(3/2) = 0,176.
- Для слова "технології": з'являється у 2 документах з 3, IDF = log(3/2) = 0,176.
- Для слів "змінює", "світ", "розвиваються", "штучному", "інтелекту": з'являються по одному разу, IDF = log(3/1) = 0.477.

Розрахунок TF-IDF:

- Для слова "штучний" у документі 1: TF = 1/4 = 0,25; TF-IDF = 0,25 × 0,176 = 0,044.
- Для слова "інтелект" у документі 1: TF = 1/4 = 0,25; TF-IDF = 0,25 × 0,176 = 0,044.
- Для слова "змінює" в документі 1: TF = 1/4 = 0,25; TF-IDF = 0,25 × 0,477 = 0,119.
- Для слова "світ" у документі 1: TF = 1/4 = 0,25; TF-IDF = 0,25×0,477 = 0,119.

Цей приклад показує, як TF-IDF допомагає визначити важливість термінів у контексті документа, одночасно знижуючи вагу загальних слів і підвищуючи вагу рідкісних і значущих слів.

ISSN 2788-6603

- *Приклад застосування моделі PageRank*

Розглянемо простий приклад, щоб продемонструвати, як працює алгоритм PageRank. Припустимо, у нас є невелика мережа вебсторінок: *A*, *B*, *C*, і *D*, з такими посиланнями між ними:

- Сторінка A посилається на сторінку B.
- Сторінка B посилається на сторінки A і C.
- Сторінка C посилається на сторінку B.
- Сторінка D посилається на сторінки C і A.

Початкова ініціалізація

Припустимо, що ми маємо 4 сторінки. Спочатку всім сторінкам призначають однакове значення PageRank:

$$PR(A) = PR(B) = PR(C) = PR(D) = \frac{1}{4} = 0{,}25.$$

Для простоти, використовуємо коефіцієнт затухання d = 0,85.

Обчислення PageRank для сторінки A

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(D)}{L(D)}\right),$$

$$PR(A) = \frac{1-0{,}85}{4} + 0{,}85\left(\frac{0{,}25}{2} + \frac{0{,}25}{2}\right) = 0{,}25\ .$$

Обчислення PageRank для сторінки B

$$PR(B) = \frac{1-d}{N} + d\left(\frac{PR(A)}{L(A)} + \frac{PR(C)}{L(C)}\right),$$

$$PR(B) = \frac{1-0{,}85}{4} + 0{,}85\left(\frac{0{,}25}{1} + \frac{0{,}25}{1}\right) = 0{,}4625.$$

Обчислення PageRank для сторінки C

$$PR(C) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)}\right),$$

$$PR(C) = \frac{1-0{,}85}{4} + 0{,}85\left(\frac{0{,}25}{2}\right) = 0{,}14375.$$

Обчислення PageRank для сторінки D

$$PR(D) = \frac{1-d}{N} + d\left(\frac{PR(C)}{L(C)}\right),$$

$$PR(D) = \frac{1-0{,}85}{4} + 0{,}85\left(\frac{0{,}25}{1}\right) = 0{,}25.$$

Алгоритм PageRank вимагає багаторазових ітерацій для досягнення стабільності (збіжності). На практиці цей процес триває, поки зміни у значеннях PageRank між ітераціями не стають дуже малими.

Після декількох ітерацій (зазвичай десятків або сотень) алгоритм збігається, і ми отримуємо стабільні значення PageRank для всіх сторінок. У реальних сценаріях для великих мереж вебсторінок обчислення PageRank можуть бути складнішими, але основні принципи залишаються такими самими.

- *Приклад застосування TextRank*

Щоб продемонструвати роботу TextRank, розглянемо приклад автоматичного реферування тексту. Припустимо, у нас є такий текст:

"Сонце яскраво світило над морем, і хвилі спокійно накочувалися на берег. Птахи співали в гілках дерев, а вітер ніжно шелестів у листі. Мандрівник стояв на вершині скелі і дивився на простори, які розкинулися перед ним. Він відчував себе вільним і щасливим, адже дорога привела його до місця, де він завжди мріяв побувати. Спокій і краса цього місця заполонили його душу".

Кроки роботи TextRank

1. Побудова графа:
   - Кожне речення у тексті представляється як вузол у графі.
   - Зв'язки між реченнями визначаються на основі схожості (напр., на основі кількості спільних слів або схожих фраз).
2. Обчислення схожості між реченнями:
   - TextRank обчислює схожість між кожною парою речень (напр., якщо два речення мають спільні слова або схожі фрази, вони отримують вищий бал схожості).
3. Ранжування речень:
   - Алгоритм ітеративно обчислює "вагу" кожного речення, тобто визначає його важливість, з огляду на те, скільки інших речень із ним пов'язано та наскільки вони важливі.
4. Вибір найважливіших речень:
   - Після кількох ітерацій TextRank отримує ранжування речень. Найважливіші з них вибираються для формування реферату.

Результат та його інтерпретація. Після застосування TextRank до нашого тексту, наприклад, можуть бути відібрані такі речення:

"Мандрівник стояв на вершині скелі і дивився на простори, які розкинулися перед ним. Він відчував себе вільним і щасливим, адже дорога привела його до місця, де він завжди мріяв побувати."

TextRank вибрав саме ці речення, оскільки вони є центральними у тексті та найбільш зв'язані з іншими реченнями через ключові слова та ідеї, такі як "мандрівник", "щасливим", "мріяв", "місце". Інші речення, такі як ті, що описують природу ("Сонце яскраво світило над морем..."), були менш зв'язані із центральною ідеєю тексту, тому їх не включили до реферату.

Цей приклад показує, як TextRank автоматично відбирає найважливіші частини тексту для створення короткого реферату, зберігаючи основний зміст.

- *Приклад застосування GPT*

Уявімо, що вам потрібно створити коротку статтю про вплив штучного інтелекту на сучасну медицину. Ви можете використовувати GPT для автоматичного створення тексту на цю тему.

Запит і вихідний текст, згенерований GPT. "Напиши коротку статтю про те, як штучний інтелект змінює медицину".

Штучний інтелект: Революція в сучасній медицині. Штучний інтелект (ШІ) стає ключовим елементом у розвитку сучасної медицини, відкриваючи нові горизонти для діагностики, лікування та досліджень. Здатність ШІ аналізувати величезні обсяги даних дозволяє лікарям швидше і точніше виявляти захворювання на ранніх стадіях, що значно підвищує шанси на успішне лікування.

Одним із найяскравіших прикладів використання ШІ є розроблення алгоритмів для аналізу медичних зображень. Такі системи вже нині здатні діагностувати ракові утворення з точністю, що перевершує людські можливості. Крім того, ШІ допомагає персоналізувати лікування, створюючи індивідуальні терапевтичні плани на основі генетичних даних та історії хвороби пацієнта.

Не менш важливим є застосування ШІ у сфері фармакології. Завдяки моделям машинного навчання, час розроблення нових ліків значно скорочується, а їхня ефективність підвищується. Наприклад, алгоритми можуть прогнозувати, як різні сполуки взаємодіятимуть із людським організмом, що дозволяє уникнути багатьох побічних ефектів.

Однак, незважаючи на всі переваги, використання ШІ в медицині стикається з певними викликами. Наприклад, питання конфіденційності даних пацієнтів та етичні аспекти впровадження нових технологій потребують ретельного вивчення й регулювання.

Загалом, штучний інтелект вже нині змінює медицину, робить її точнішою, ефективнішою та доступнішою. У майбутньому ці технології стануть ще більш інтегрованими в медичну практику, забезпечуючи нові можливості для здоров'я та добробуту людей.

Аналіз прикладу

- Структура: текст має чітку структуру зі вступом, основним змістом і висновком.
- Контент: модель згадує про конкретні сфери медицини, де ШІ вже активно застосовується, такі як діагностика та фармакологія.
- Зв'язність: текст добре зв'язаний, кожен абзац логічно випливає з попереднього.
- Загальна якість: модель генерує текст, який виглядає правдоподібно і може бути корисним у реальних сценаріях.

Цей приклад демонструє, як GPT може використовуватися для автоматичного створення текстів на задану тему, зберігаючи при цьому зв'язність, логічність і структурованість тексту.

- *Приклад застосування моделі BART*

BART можна застосовувати для різних задач оброблення тексту. Розглянемо приклад використання BART для **автоматичного реферування тексту** – це процес створення короткого викладу (реферату) довгого документа.

Сценарій. Уявімо, що ви працюєте з великим документом, який містить детальний огляд досліджень у галузі зміни клімату. Потрібно створити короткий реферат цього документа, щоб швидко передати основні ідеї читачам.

"Зміна клімату стала однією з найважливіших глобальних проблем нашого часу. Вчені прогнозують, що підвищення середньої температури на планеті призведе до значних екологічних і соціальних наслідків. Танення льодовиків, підвищення рівня моря, зміни в екосистемах і погодних умовах – це лише деякі з наслідків, які вплинуть на людство. Дослідження показують, що вразливі групи населення, зокрема жителі прибережних регіонів і країн, що розвиваються, будуть найбільш вразливі до цих змін. Крім того, зміна клімату може негативно вплинути на світову економіку, сільське господарство і здоров'я людей".

Постановка завдання і результат, який був згенерований BART

Згенерувати короткий реферат цього тексту.

"Зміна клімату спричиняє серйозні екологічні та соціальні наслідки, такі як танення льодовиків, підвищення рівня моря і вплив на вразливі групи населення. Ці зміни також можуть мати негативний вплив на світову економіку і здоров'я людей".

Аналіз результату

- Стислий виклад: BART зміг виокремити ключові аспекти тексту, зокрема основні загрози, пов'язані зі зміною клімату.
- Збереження смислу: згенерований реферат зберігає основні ідеї, викладені в оригінальному тексті.
- Зв'язність: текст залишився логічним і зв'язним, що важливо для автоматичного реферування.

Такий підхід можна використовувати в новинних агенціях для створення коротких підсумків новинних статей, у наукових дослідженнях для автоматичного створення анотацій статей або в бізнесі для підготовки звітів на основі великих документів.

Переваги та недоліки цього прикладу

- Переваги: економія часу та зусиль під час створення коротких підсумків великих текстів.
- Недоліки: можлива втрата важливих деталей або генерація узагальнень, які не повністю відображають усі аспекти оригінального тексту.

Цей приклад демонструє, як BART може використовуватися для автоматичного створення коротких рефератів із довгих текстів, що є корисним у багатьох професійних і дослідницьких контекстах.

- *Приклад застосування моделі ROUGE*

Давайте додамо докладні розрахунки для кожної метрики ROUGE в наведеному прикладі.

ISSN 2788-6603

Вхідні дані

- Оригінальний текст статті: "Сьогодні вранці у центрі Києва стався потужний вибух. За попередніми даними, постраждало кілька людей. Вибух стався біля головного офісу відомої компанії. На місце події прибули рятувальники та поліція. Причини вибуху наразі з'ясовуються".
- Еталонний реферат (створений людиною): "Вибух у центрі Києва: постраждало кілька людей, причини з'ясовуються".
- Автоматичний реферат (згенерований системою): "Потужний вибух у Києві: постраждало кілька людей, причини невідомі".

Застосування ROUGE

ROUGE-1 (уніграми)

Ціль: оцінити кількість збігів окремих слів між автоматичним рефератом і еталонним.

- Слова в еталонному рефераті: "вибух", "у", "центрі", "Києва", "постраждало", "кілька", "людей", "причини", "з'ясовуються".
- Слова в автоматичному рефераті: "потужний", "вибух", "у", "Києві", "постраждало", "кілька", "людей", "причини", "невідомі".

Спільні слова між еталонним і автоматичним рефератами:

- "Вибух", "у", "постраждало", "кілька", "людей", "причини".

Розрахунок Precision (точність):

- Спільні уніграми: 6.
- Загальна кількість уніграмів в автоматичному рефераті: 9.
- ROUGE-1 Precision = 6/9 ≈ 0,67.

Розрахунок Recall (повнота):

- Спільні уніграми: 6.
- Загальна кількість уніграмів в еталонному рефераті: 9.
- ROUGE-1 Recall = 6/9 ≈ 0.67.

Розрахунок F1-міри (Christen, Hand, & Kirielle, 2024):

- ROUGE-1 F1 = 2 × (Precision × Recall) / (Precision + Recall).
- ROUGE-1 F1 ≈ 2 × (0,67 * 0,67) / (0,67 + 0,67) ≈ 0,67.

ROUGE-2 (біграми)

Ціль: оцінити кількість збігів пар слів (біграм) між автоматичним рефератом і еталонним.

- Біграми в еталонному рефераті: "вибух у", "у центрі", "центр Києва", "Києва постраждало", "постраждало кілька", "кілька людей", "людей причини", "причини з'ясовуються".
- Біграми в автоматичному рефераті: "потужний вибух", "вибух у", "у Києві", "Києві постраждало", "постраждало кілька", "кілька людей", "людей причини", "причини невідомі".

Спільні біграми між еталонним і автоматичним рефератами:

- "Вибух у", "постраждало кілька", "кілька людей", "людей причини".

Розрахунок Precision (точність):

- Спільні біграми: 4.
- Загальна кількість біграмів в автоматичному рефераті: 8.
- ROUGE-2 Precision = 4/8 = 0,50.

Розрахунок Recall (повнота):

- Спільні біграми: 4.
- Загальна кількість біграм в еталонному рефераті: 8.
- ROUGE-2 Recall = 4/8 = 0,50.

Розрахунок F1-міри:

- ROUGE-2 F1 = 2 * (Precision × Recall) / (Precision + Recall).
- ROUGE-2 F1 = 2 * (0,50 × 0,50) / (0,50 + 0,50) = 0,50.

ROUGE-L (довгі спільні підпослідовності)

Ціль: Оцінити довжину найдовшої спільної підпослідовності слів, що зберігає порядок слів.

- Еталонний реферат: "Вибух у центрі Києва: постраждало кілька людей, причини з'ясовуються".
- Автоматичний реферат: "Потужний вибух у Києві: постраждало кілька людей, причини невідомі".

Найдовша спільна підпослідовність:

- "Вибух у Києві постраждало кілька людей причини".

Розрахунок Precision (точність):

- Довжина спільної підпослідовності: 7 слів.
- Загальна кількість слів в автоматичному рефераті: 9.
- ROUGE-L Precision = 7/9 ≈ 0,78.

Розрахунок Recall (повнота):

- Довжина спільної підпослідовності: 7 слів.
- Загальна кількість слів в еталонному рефераті: 9.
- ROUGE-L Recall = 7/9 ≈ 0,78.

Розрахунок F1-міри:

- ROUGE-L F1 = 2 × (Precision × Recall) / (Precision + Recall).
- ROUGE-L F1 ≈ 0,78.

- *Приклад застосування моделі BLEU*

Наведемо приклад застосування метрики BLEU для оцінювання якості машинного перекладу з розрахунками.

Уявімо, що маємо еталонний переклад і кандидатський переклад від системи машинного перекладу. Ми хочемо оцінити якість кандидатського перекладу за допомогою метрики BLEU.

Еталонний переклад: "The quick brown fox jumps over the lazy dog".

Кандидатський переклад: "The fast brown fox jumps over the lazy dog".

Розрахунок n-грам

Метрика BLEU порівнює *n*-грам кандидатського перекладу з *n*-грамами еталонного перекладу. Давайте почнемо з розрахунку 1-грам (одиночних слів), а потім перейдемо до 2-грам (пари слів).

1-грам:

- Еталонний переклад: ["The", "quick", "brown", "fox", "jumps", "over", "the", "lazy", "dog"].
- Кандидатський переклад: ["The", "fast", "brown", "fox", "jumps", "over", "the", "lazy", "dog"].

2-грам:

- Еталонний переклад: ["The quick", "quick brown", "brown fox", "fox jumps", "jumps over", "over the", "the lazy", "lazy dog"].
- Кандидатський переклад: ["The fast", "fast brown", "brown fox", "fox jumps", "jumps over", "over the", "the lazy", "lazy dog"].

Підрахунок збігів n-грам

1-грам:

- Кількість збігів: ["The", "brown", "fox", "jumps", "over", "the", "lazy", "dog"] = 8 збігів із 9 (тут "fast" не збігається з "quick").
- Загальна кількість 1-грам у кандидатському перекладі: 9.

2-грам:

- Кількість збігів: ["brown fox", "fox jumps", "jumps over", "over the", "the lazy", "lazy dog"] = 6 збігів із 8.
- Загальна кількість 2-грам у кандидатському перекладі: 8.

Обчислення точності для n-грам

1-грамна точність (p_1):

$$p_1 = \frac{\textit{Кількість збігів 1-грам}}{\textit{Загальна кількість 1-грам у кандидаті}} = \frac{8}{9} \approx 0{,}889.$$

2-грамна точність (p_2):

$$p_2 = \frac{\textit{Кількість збігів 2-грам}}{\textit{Загальна кількість 2-грам у кандидаті}} = \frac{6}{8} = 0{,}75.$$

Обчислення середньої геометричної точності

Метрика BLEU зазвичай використовує середню геометричну точність для різних *n*-грам. Для простоти візьмемо тільки 1-грамну і 2-грамну точності:

$$\textit{Середня геометрична точність} = \sqrt{p_1 \times p_2} = \sqrt{0{,}889 \times 0{,}75} \approx 0{,}816.$$

Штраф за кратність (Brevity Penalty)

BLEU включає штраф за кратність для запобігання надмірно коротким перекладам. Якщо довжина кандидатського перекладу менша за довжину еталонного, застосовується штраф.

В нашому випадку, довжина кандидатського перекладу (9 слів) така сама, як і довжина еталонного перекладу (9 слів), тому штраф за кратність дорівнює 1.

Обчислення BLEU

Остаточна формула BLEU включає обчислену середню геометричну точність і штраф за кратність:

BLEU = Штраф за кратність * Середня геометрична точність.

Оскільки штраф за кратність дорівнює 1:

BLEU = 1 × 0,816 ≈ 0,816.

Отже, у цьому прикладі BLEU оцінка для кандидатського перекладу становить приблизно 0,816 або 81,6 %, що свідчить про високу схожість кандидатського перекладу з еталонним. Цей результат показує, що переклад є якісним, хоча є деякі незначні відмінності, такі як заміна "quick" на "fast".

Результати

Результати дослідження свідчать, що метрики ROUGE показують хорошу точність у вимірюванні збігів *n*-грам (послідовностей з *n* слів), тоді як BLEU ефективна у завданнях машинного перекладу, але може не враховувати деякі синтаксичні особливості тексту. Оцінювання методів автоматичного реферування за допомогою цих метрик показало, що екстрактивні методи реферування, такі як TF-IDF, є ефективними для оброблення простих текстів, але можуть втратити важливий контекст у складних текстах. PageRank і TextRank дозволяють враховувати зв'язки між реченнями, проте можуть давати менш релевантні результати для текстів із слабко вираженими структурними зв'язками. Абстрактні моделі GPT і BART забезпечують гнучкіший підхід до реферування, створюючи нові речення, що краще передають зміст, однак потребують значних обчислювальних ресурсів і складні у впровадженні.

Дискусія і висновки

Порівняльний аналіз результатів за допомогою метрик ROUGE і BLEU показав, що кожен підхід має свої сильні та слабкі сторони. Метрика ROUGE добре підходить для оцінювання екстрактивних методів, де важливо враховувати збіг *n*-грам з еталонним текстом. BLEU, зі свого боку, ефективна у задачах, де важливо зберігати структурні особливості перекладу або переказу, проте її застосування в абстрактних методах може бути обмеженим через недостатнє врахування контексту та синонімії.

Класичні підходи, такі як екстрактивне реферування, використовують різні статистичні та лінгвістичні методи для вибору найбільш значущих речень із тексту. Хоча саме класичні добре підходять для формальних і структурованих текстів, вони мають обмеження в передачі сенсу, особливо для текстів із складними взаємозв'язками між реченнями.

ISSN 2788-6603

TF-IDF є ефективним інструментом для виділення ключових термінів у тексті на основі їхньої частоти й унікальності. Однак цей метод не враховує контекст і не працює добре із синонімами, що може призводити до втрати сенсу у виділених реченнях.

PageRank є потужним інструментом для ранжування елементів на основі зв'язків між ними. У текстовому аналізі він допомагає визначити важливість речень через взаємозв'язки, але може мати проблеми з контекстуальним розумінням інформації, що іноді призводить до неправильних результатів.

Методи на основі графів, такі як TextRank, ефективно використовують структурні зв'язки в тексті для виявлення важливих елементів. Вони є гнучкими та підходять для багатьох типів текстів, проте вразливі до слабких зв'язків і можуть пропускати важливу інформацію в текстах зі складною структурою.

Абстрактні методи надають можливість створювати нові речення, що відображають загальний зміст тексту. Це робить їх гнучкішими порівняно з екстрактивними методами, але вони значно складніші у впровадженні та потребують великих обсягів навчальних даних для досягнення високої якості результату.

GPT є однією із провідних моделей для абстрактного реферування завдяки своїй здатності генерувати текст, що відображає глибокий зміст і контекст. Переваги включають високу якість і креативність генерованих текстів, але модель може іноді генерувати неправдиву або несуттєву інформацію, що є її основним недоліком.

BART поєднує переваги двох підходів: двонаправленого й авторегресивного, що дозволяє досягати високої точності в завданнях текстового реферування. Він особливо добре працює зі складними текстами, але вимагає значних обчислювальних ресурсів і обсягу навчальних даних.

Для оцінювання якості рефератів використовують кількісні метрики – ROUGE і BLEU, які забезпечують об'єктивні критерії для порівняння кандидатських рефератів з еталонними. Однак ці метрики мають свої обмеження, пов'язані з нездатністю повністю врахувати семантику і контекст тексту.

ROUGE є основною метрикою для оцінювання рефератів, яка враховує збіги *n*-грам між кандидатським і еталонним текстом. Це робить її корисною для оцінювання якості рефератів, але метрика погано враховує синоніми і варіативність у формулюваннях, що може знижувати її точність.

BLEU є потужною метрикою для оцінювання якості машинного перекладу, але її застосування обмежене нездатністю враховувати синоніми, контекст і граматику. Її результати слід інтерпретувати з урахуванням цих обмежень і, за можливості, доповнювати іншими методами оцінювання.

На основі проведеного аналізу можна надати такі рекомендації.

1. **Використання екстрактивних методів** доцільно для задач, де важлива швидкість оброблення текстів і коли тексти мають чітку структуру. Ці методи особливо корисні для оброблення великих обсягів даних, де необхідна простота реалізації та мінімальні вимоги до ресурсів.
2. **Абстрактні методи** є придатнішими для задач, де необхідна висока точність передачі змісту та контексту тексту. Вони рекомендовані для застосування в умовах, де є доступ до значних обчислювальних потужностей і де якість рефератів має пріоритетне значення.
3. **Поєднання обох підходів** може бути оптимальним рішенням для створення гібридних систем реферування, які можуть використовувати переваги екстрактивних і абстрактних методів залежно від специфіки текстів і вимог до кінцевого результату.

Отже, вибір методу для автоматичного реферування текстів повинен базуватися на конкретних вимогах до точності, швидкості та ресурсів, а також на особливостях самих текстів, що обробляються. Поєднання різних підходів і адаптація моделей під конкретні задачі дозволить отримати найкращі результати у створенні якісних і релевантних рефератів.

Внесок авторів: Олексій Кузнєцов – дослідження й аналіз методів та метрик, використаних у статті, написання частини статті; Геннадій Кисельов – огляд літературних джерел, розроблення висновків і рекомендацій, координація роботи, написання частини статті.

Список використаних джерел

Кузнєцов, О., & Кисельов, Г. (2022). Методи розпізнавання текстів та пошуку ключових слів для автоматичного реферування текстів. *Системні науки та інформатика: збірник доповідей І науково-практичної конференції "Системні науки та інформатика", 22–29 листопада 2022 року* (с. 331–335). Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського". https://ai.kpi.ua/ua/document2022.pdf

Callison-Burch, C., Osborne, M., & Koehn, P. (2006). Re-evaluating the Role of Bleu in Machine Translation Research. *In 11th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 249–256). Trento, Italy. Association for Computational Linguistics. https://aclanthology.org/E06-1032/

Christen, P., J. Hand, D., & Kirielle, N. (2024). A review of the F-measure: Its History, Properties, Criti-cism, and Alternatives. *ACM Computing Surveys*, 56(3), 1–24. https://doi.org/10.1145/3606367

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 (Long and Short Papers), (pp. 4171–4186). Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Hosna, A., Merry, E., Gyalmo, J., Alom, Z, Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal of Big Data,* 9(102). https://doi.org/10.1186/s40537-022-00652-w

Kuznietsov, O., & Kyselov, G. (2024). An overview of current issues in automatic text summarization of natural language using artificial intelligence methods. *Technology Audit and Production Reserves*, 4(78), 12–19. https://journals.uran.ua/tarp/article/view/309472

Lewis, M., Liu,Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* online (pp. 7871–7880). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out (pp. 74–81). Barcelona, Spain. Association for Computational Linguistics. https://aclanthology.org/W04-1013.pdf

Lin, C.-Y., & Hovy, E. H. (2000). The Automated acquisition of topic signatures for text summarization. In *Proceedings of COLING-00*. Saarbrücken, Germany (pp. 495–501). International Committee on Computational Linguistics. https://doi.org/10.1007/978-3-540-74851-9_25

Mihalcea, R., & Tarau P. (2004). *TextRank: Bringing order into texts.* Association for Computational Linguistics. https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf

Moratanch, N., & Chitrakala, S. (2016). A survey on abstractive text summarization, 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) (pp. 1–7). Nagercoil, India. Institute of Electrical and Electronics Engineers. https://ieeexplore.ieee.org/document/7530193

OpenAI. (2022). ChatGPT. https://openai.com/chatgpt/

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, (pp. 311–318). Association for Computational Linguistics. http://aclweb.org/anthology/P/P02/P02-1040.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, Aidan, N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems,* 30. Curran Associates, Inc. https://arxiv.org/abs/1706.03762

References

Callison-Burch, C., Osborne, M., Koehn, P. (2006). Re-evaluating the Role of Bleu in Machine Translation Research. In 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy. Association for Computational Linguistics (pp. 249–256). https://aclanthology.org/E06-1032/

Christen, P., J. Hand, D., & Kirielle, N. (2024). A review of the *F*-measure: Its History, Properties, Criticism, and Alternatives. *ACM Computing Surveys*, 56(3), 1–24. https://doi.org/10.1145/3606367

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota. Association for Computational Linguistics, (pp. 4171–4186). https://doi.org/10.18653/v1/N19-1423

Hosna, A., Merry, E., Gyalmo, J., Alom, Z, Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal of Big Data,* 9(102). https://journals.uran.ua/tarp/article/view/309472

Kuznetsov, O., & Kyselyov, G. (2022). Methods of Text Recognition and Keyword Search for Automatic Text Summarization. System Sciences and Informatics: *Proceedings of the 1st Scientific and Practical Conference "System Sciences and Informatics", November 22–29, 2022* (pp. 331–335). National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" [in Ukrainian]. https://ai.kpi.ua/ua/document2022.pdf

Kuznietsov, O., & Kyselov, G. (2024). An overview of current issues in automatic text summarization of natural language using artificial intelligence methods. Technology Audit and Production Reserves, 4(78), 12–19. https://journals.uran.ua/tarp/article/view/309472

Lewis, M., Liu,Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online. (pp. 7871–7880). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out (pp. 74–81). Barcelona, Spain. Association for Computational Linguistics. https://aclanthology.org/W04-1013.pdf

Lin C.-Y., & Hovy E.H. (2000) The Automated acquisition of topic signatures for text summarization. In Proceedings of COLING-00. Saarbrücken, Germany. (pp. 495-501). International Committee on Computational Linguistics. https://doi.org/10.1007/978-3-540-74851-9_25

Mihalcea, R., & Tarau P. (2004). TextRank: Bringing order into texts. Association for Computational Linguistics. https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf

Moratanch, N, & Chitrakala, S. (2016). A survey on abstractive text summarization, 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Nagercoil, India, 2016, (pp. 1–7). Institute of Electrical and Electronics Engineers. https://ieeexplore.ieee.org/document/7530193/

OpenAI. (2022). ChatGPT. https://openai.com/chatgpt/

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, (pp. 311–318). Association for Computational Linguistics. http://aclweb.org/anthology/P/P02/P02-1040.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, Aidan, N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems,* 30. Curran Associates, Inc. https://arxiv.org/abs/1706.03762

Отримано редакцією журналу / Received: 15.08.24
Прорецензовано / Revised: 10.09.24
Схвалено до друку / Accepted: 22.09.24

Oleksii KUZNIETSOV, PhD Student
ORCID ID: 0000-0002-3537-9976
e-mail: oleksiy1908@gmail.com
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Gennadiy KYSELOV, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0003-2682-3593
e-mail: g.kyselov@gmail.com
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

USING AND ANALYSIS OF FORMAL METHODS FOR EVALUATING THE RELEVANCE OF AUTOMATICALLY GENERATED SUMMARIES OF INFORMATIONAL TEXTS

Background. ***The article reviews existing approaches to evaluating the quality of automatically generated summaries of informational texts. It provides an overview of automatic summarization methods, including classical approaches and modern models based on artificial intelligence. The review covers extractive summarization methods such as TF-IDF and PageRank, as well as graph-based methods, specifically TextRank. Special attention is given to abstractive approaches, including Generative Pretrained Transformer (GPT) and Bidirectional and Auto-Regressive Transformers (BART) models. The quality of generated summaries is evaluated using quantitative metrics of summary relevance, particularly ROUGE and BLEU.***

Methods. ***The article analyzes several approaches to automatic text summarization. Classical extractive methods, such as TF-IDF, calculate the importance of terms based on their frequency within a document and across a collection of documents. PageRank and TextRank utilize graph models to determine the significance of sentences based on the connections between them. Abstractive methods, such as GPT and BART, generate new sentences that succinctly convey the content of the original text. The effectiveness of each approach is assessed using ROUGE and BLEU metrics, which measure the overlap between automatically generated summaries and reference texts. Particular attention is given to analyzing their accuracy, flexibility, resource requirements, and ease of implementation.***

Results. ***The results of the study show that ROUGE metrics demonstrate good accuracy in measuring n-gram overlaps (sequences of n words), while BLEU is effective in machine translation tasks but may not account for certain syntactic features of the text. The evaluation of automatic summarization methods using these metrics revealed that extractive summarization methods, such as TF-IDF, are effective for processing simple texts but may lose important context in complex texts. PageRank and TextRank consider the connections between sentences but may produce less relevant results for texts with weak structural connections. Abstractive models like GPT and BART provide a more flexible approach to summarization, creating new sentences that better convey the meaning, though they require significant computational resources and are complex to implement.***

Conclusions. ***Combining classical and modern methods of automatic text summarization allows for achieving higher quality results. It is important to consider the specificity of the text and the requirements for the final outcome, adapting the selected approaches and metrics according to the task.***

Keywords: ***automatic summarization, extractive methods, abstractive methods, GPT, BART, ROUGE, BLEU, TextRank, PageRank, TF-IDF.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

UDC 004.94
DOI: https://doi.org/10.17721/AIT.2024.1.05

Vitalii OMELCHENKO, PhD Student
ORCID ID: 0000-0002-3850-6555
e-mail: vitaly.om25@gmail.com
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Oleksandr ROLIK, DSc (Engin.), Prof.
ORCID ID: 0000-0001-8829-4645
e-mail: arolick@gmail.com
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

HYBRID METHOD FOR HORIZONTAL AND VERTICAL COMPUTATIONAL RESOURCE SCALING

Background. ***Increasing the efficiency of cluster computing resources while maintaining the established QoS levels is a critical task in IT infrastructure management. Dynamic management of computing resources, in particular vertical and horizontal scaling, are tools that allow automating the processes of adapting applications to dynamic loads. The aim of the work is to improve the efficiency of existing scaling methods by combining them.***

Methods. ***A hybrid scaling method using the coordination module is proposed. This module coordinates the operation of vertical and horizontal scaling components based on the given constraints, priorities, and the current state of the system. The coordination module aims to increase efficiency of the both components and prevents inconsistency in the combination of the number of instances and resource requests for each instance.***

Results. ***To achieve the given objective, a hybrid method of vertical and horizontal scaling using priority-based component coordination was developed. Priority configuration affects the order of components operation. In the case of horizontal-vertical order, the non-prior vertical component does not affect configuration of the prior horizontal component. The developed method is evaluated based on modeling the operation of an application with a load containing a constant seasonality. The experiments demonstrate a 65% reduction in the unprofitable reservation of cluster computing resources compared to static requests.***

Conclusions. ***The developed method can be used to increase the efficiency of resource utilization in clusters under dynamic loads compared to basic scaling methods. In further research, it is necessary to evaluate the method using real infrastructure. It is also necessary to investigate the work of a hybrid method using a predictive approach.***

Keywords: ***information technologies, information systems, resource management, scaling, IT-infrastructure, vertical scaling, horizontal scaling.***

Background

The dynamic allocation of computational resources for cloud applications enables the delivery of the agreed level of quality of service (QoS) to customers. An increasing number of enterprise informational systems (IS) rely on existing solutions for the management of computational resources (Rolik, Telenik, & Yasochka, 2018).

Horizontal and vertical scaling of applications in a cluster is one of the most effective tools for automating these processes in clusters (Lorido-Botran, Miguel-Alonso, & Lozano, 2014). Horizontal scaling is more versatile than vertical scaling because it provides fault tolerance, is not limited to the resources of a single physical machine, and allows flexible management of the amount of resources allocated (Al-Dhuraibi et al., 2017). In practice, vertical scaling is used when an application cannot be distributed, such as databases (Omelchenko, & Rolik, 2022). Although horizontal scaling is a flexible way of dynamically allocating resources, this process takes place within the limits of defined requirements. If the amount of computing resources allocated to the application and the actual load on each resource are not balanced, then part of this resource is not used. By adjusting the volume of requests within an application, its instances can be placed on more specialized VMs, reducing financial costs while maintaining the same level of QoS (Rochman, Levy, & Brosh, 2014).

The limitations of standalone scaling methods and the high demands on modern IS resulted in the emergence of hybrid methods (Straesser et al., 2022). In particular, vertical-horizontal and reactive-proactive methods combine the advantages of existing approaches to improve overall performance (Singh et al., 2019). However, the main disadvantage of hybrid methods is the complexity of coordinating individual components.

This work aims to increase the efficiency of computing resource utilization while maintaining the established level of QoS in dynamic resource management. To achieve this goal, a combined scaling method is proposed, which includes coordination of horizontal and vertical scaling components.

Problem statement. Horizontal scaling is the process of increasing or decreasing the number of application instances to ensure QoS levels within the agreed SLA and to increase the efficiency of the use of the cluster's compute resources (Rolik, & Omelchenko, 2024). This type of scaling can work with either a single resource type or multiple resources. In the case of scaling a single resource at a time, the scaling of the *l*-th application can be described as follows:

$$k_l(t) = \left\lceil \frac{W_l(t)}{X_l} \right\rceil, l = 1, ..., m,$$

where m is the number of applications, $W_l(t)$ is the utilization of the target compute resource, X_l is the request for the target resource type. This is the level of utilization of other types of computing resources, which can lead to both denial of service and significant unprofitable redundancy (Omelchenko, & Rolik, 2022).

In the case of scaling using a set of resources, the one with the highest utilization percentage in relation to the target value is selected:

$$k_l(t) = \max\left(\left\lceil \frac{W_{ly}(t)}{X_{ly}} \right\rceil \right), l = 1, ..., m, y = 1, ..., h,$$

where m is the number of applications, $W_{ly}(t)$ is the usage of the y-th computing resource, and X_{ly} is the request for the y-th type of resource. Accordingly, only one of the resources is taken into account and the others may be unbalanced.

The horizontal scaling model described has the disadvantage of insufficient utilization of non-priority resource types. A non-priority resource does not affect the amount of resources allocated for scheduling in general or at any particular time (Sedaghat, Hernandez-Rodriguez, & Elmroth, 2013). Fig. 1 shows an example of unbalanced resource allocation for an application where CPU time utilization is kept at a configured level of 90% but other resource utilization is relatively low. Reducing the number of instances is impossible due to the need to ensure a given level of QoS, and partial adjustment of the application's resource volume cannot be performed by the horizontal scaling component. To solve this problem, it is necessary to manually adjust the configuration of requests and horizontal scaling to current needs, which is a difficult task in large IS.

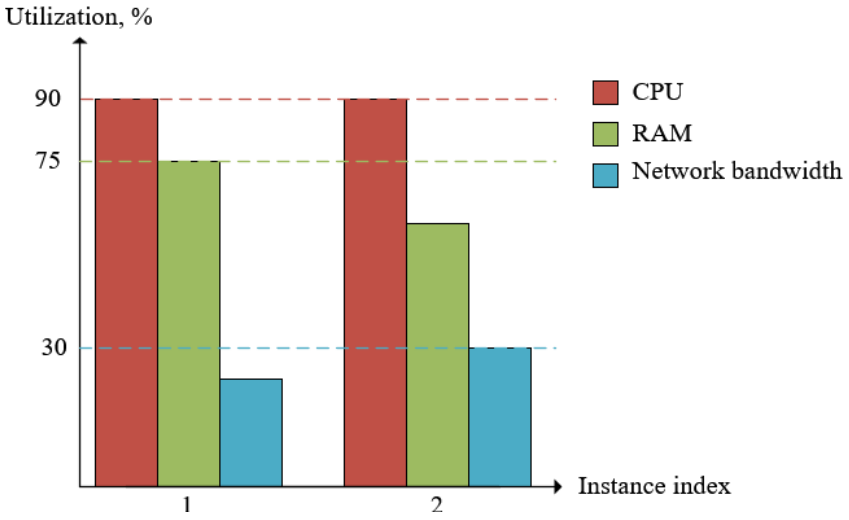


Fig. 1. Unbalanced utilization of computing resources during horizontal scaling

This paper proposes a hybrid scaling method to automate this process, which is derived from horizontal scaling but is more efficient in terms of computing resources utilization. The proposed method is designed to solve the following problems when operating horizontally scaled applications:

- increasing the level of utilization of resource types that are not a priority for horizontal scaling, in order to increase the efficiency of their use;
- resolving situations where the current instance usage of one of the resource types exceeds the set request for that resource type, in particular out-of-memory (OOM) errors leading to denial of service.

Methods

The topic of combining vertical and horizontal scaling is increasingly appearing in scientific papers (Qu, Calheiros, & Buyya, 2018). Most approaches are based on forecasting; in the paper (Dutta et al., 2012) an approach using predictive model management is proposed. The paper proposes a model with a given number of parameters that are determined based on the cluster capacity, the predicted volume of the task queue, and the application's request for computing resources. Based on the constraints and predictions, the task of optimizing resource allocation is solved. The QoS metrics of the combined method obtained as a result of experiments show an increase compared to horizontal scaling, but the evaluation of the efficiency of the use of computing resources is not examined. In the next paper (Incerto, Tribastone, & Trubiani, 2018), another approach is proposed in which the virtual machine is scaled vertically until the limit of the physical machine is met, then horizontal scaling is applied. In this work, it is assumed that horizontal scaling can provide the required level of QoS, so considerable attention is paid to the efficiency of using the cluster's computing resources. In addition, the proposed method can use both historical data and predictive models in its work. In the paper (Quattrocchi et al., 2024), the authors propose a system model based on load and response time data using queueing theory. Using the built model of the system, the solution performs an optimization task to determine the minimum amount of resources (e.g., CPU) required to provide the target response time under the current load. Another work (Millnert, & Eker, 2020) proposes a novel approach based on control theory. The authors propose using "feedback" vertical controller and "feedforward" horizontal controllers that are coordinated by calculated time delays for applying control signals.

For a successful co-operation of horizontal and vertical scaling, it is necessary to coordinate their work (Calheiros et al., 2012). The coordination should be such that the changes made by each component are fully consistent. Unprocessed conflicting changes can lead to QoS degradation and denial of service. Fig. 2 shows the coordination module containing data about the characteristics and capabilities of each scaling method.

In the coordination module, it is proposed to use the principle of priority to coordinate the work of the components when one of the components is secondary and should in no way affect the current decisions of the higher priority control component, but only adapt to the decisions provided. Uncoordinated decision making can lead to incorrect scaling of the application and a drop in QoS. The coordination module ensures that the configurations of the scaling components are compatible as follows:

$$S(x) = \begin{cases} H(x, V(x, null)), P_V > P_H, \\ V(x, H(v, null)), P_H > P_V, \end{cases}$$

where $S(x)$ is the scaling function, $H(x,c)$ and $V(x,c)$ are the scaling functions of the horizontal and vertical components respectively, taking as input the historical data and the current configuration, P_H and P_V are the priorities of the respective components.

ISSN 2788-6603

The coordination module prefers horizontal scaling, as this type is not limited by the resources of a single physical machine and is more flexible for use in modern microservices architectures. Fig. 3 demonstrates the comparison of the vertical and horizontal scaling methods.

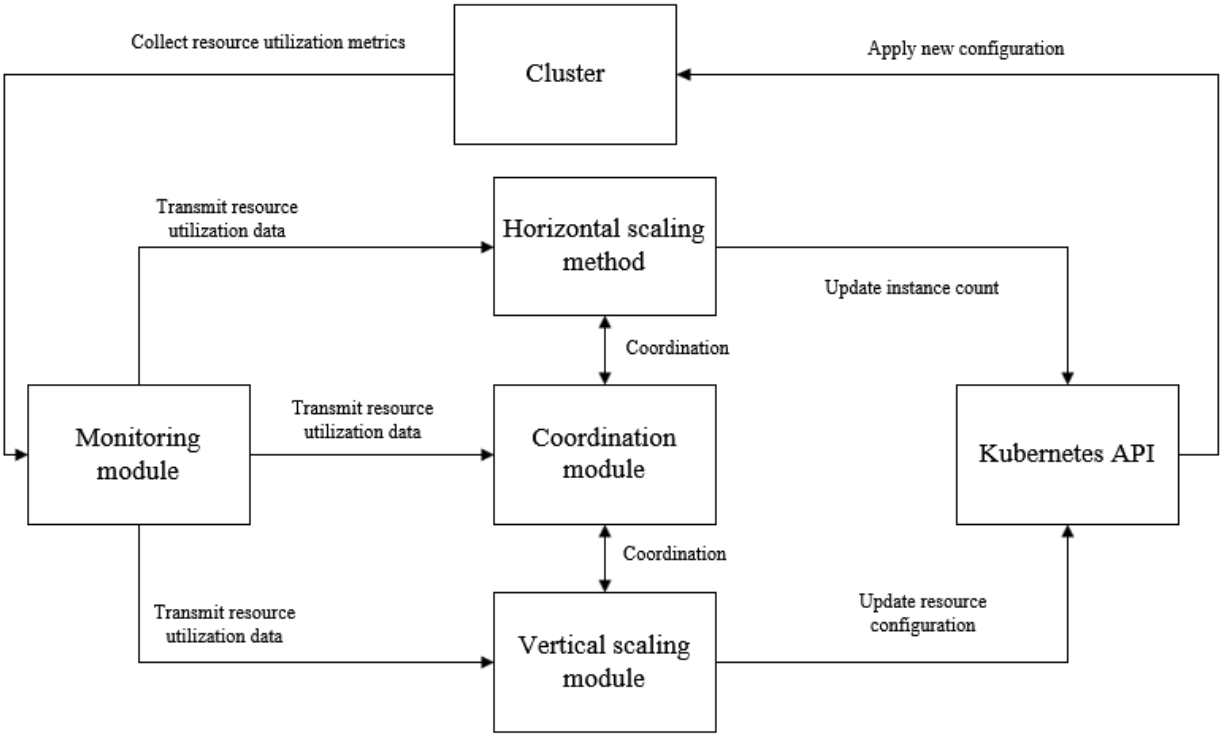


Fig. 2. Component diagram of the hybrid scaling method

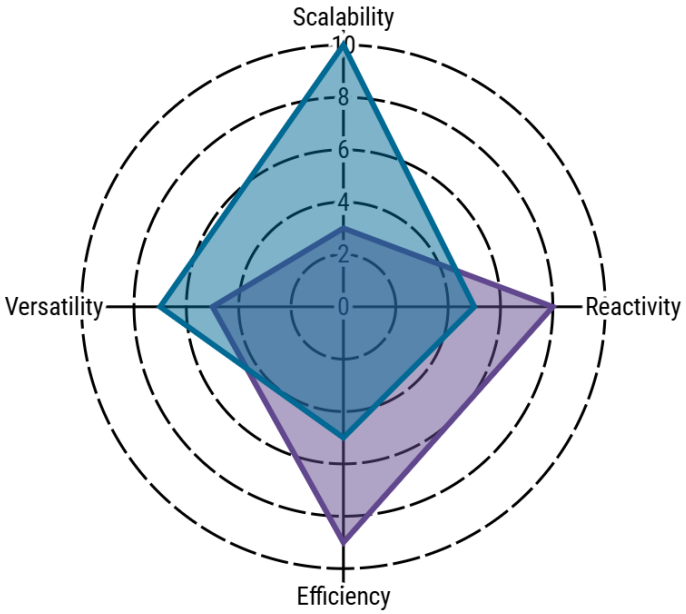


Fig. 3. Comparison of characteristics of scaling types

To implement this method, both sources of historical data on application utilization and data obtained as a result of forecasting models can be used. The data is transferred to the horizontal management module, where a decision is made on whether to increase or decrease the number of instances. This module also receives information about current resource requests $\hat{X}_{ly}$, $l = 1,\ldots,m, y = 1,\ldots,h$, where h – the number of resource types, m – the number of applications, and T_{ly}, $0 < T_{ly} \leq 1$, is the target value of utilization for each resource type. The output of the module operation is the number of instances of the l-th application k_l. An example of such a module is HPA.

In the vertical scaling module, the first step is to determine the level of utilization of each of the computing resources:

$$U_{ly} = \frac{X_{ly}}{\hat{X}_{ly}}, l = 1,...,m, y = 1,...,h,$$

where U_{ly} – utilization of the l-th application of the y-th resource type. Using the set level of target utilization T_{ly} priority resource type is found, which is used for horizontal scaling calculations:

$$y_{pivot} = \arg\max(\frac{U_{ly}}{T_{ly}}), l = 1,...,m, y = 1,...,h,$$

where y_{pivot} is the index of the resource type with the maximum relative utilization of the l-th application of the y-th type of computing resource. The next step is to estimate the required quantities of non-priority resources using the target utilization levels T_{ly}:

$$\begin{cases} \hat{X}_{ly} = f(\{X_{ly}(t_0),\ X_{ly}(t_1)...,\ X_{ly}(t_n)\}, T_{ly}),\ l = 1...m,\ y = 1...h,\ n = 1...N, \\ y \neq y_{pivot}, \end{cases}$$

where $\hat{X}_{ly}$ is the optimal values of requests for computing resources at the current iteration calculated on the basis of historical data or forecasts, for $X_{ly}(t)$ there is a transition from a scalar value to a function, since to calculate requests, it is necessary to take into account a segment of historical data or obtained forecasts, and $f(x,y)$ is a function for calculating requests. Depending on the configuration, different optimizations can be applied to this function.

When scaling using all available computing resources, it is proposed to use a generic function, the result of which guarantees the correct operation of applications and scaling components:

$$f(x,u) = \frac{\max(x)}{u},$$

where x is a vector of values for the utilization of a computing resource, and u is a target value of its utilization. If y_{pivot} is a fixed value, there is potential for optimization based on the resource type.

Each type of computing resource has its own management specifics. In this paper we propose to divide them into two groups. The first group includes resource types whose excessive usage does not lead to a complete denial of service, as the shortage is amortized in subsequent application cycles (Rodriguez, & Buyya, 2018). These types of computing resources can be artificially limited to meet defined requirements. For example, in the case of a CPU resource, the application is only given the specified amount of time to operate on the processor core, and any operations that exceed this time are performed in the next period of operation (Al-Haidari, Sqalli, & Salah, 2013). A network resource has a similar operating flow, where a queue of packets is created before transmission. For computing resources from the first group, it is proposed to use a set percentile when calculating requests:

$$f_1(x, p) = x_{sorted}\left[\left[\frac{p}{100}\cdot(n-1)\right]\right],$$

where x is a vector of values of the utilization of a certain computing resource, n is the length of the vector x, and p is target utilization percentile. This optimization can be useful for cases with short term peaks in application workload. The percentile allows you to calculate requests that cover most of the present workload scenarios and avoid resource over-provisioning.

Fig. 4 shows the integration of hybrid scaling components in a Kubernetes cluster. The monitoring components metrics-service and Prometheus provide data for horizontal scaling. The usage either predictive or reactive scaling methods is determined in the decision module. After the planned number of instances is determined, this data is passed to the vertical scaling component, which optimizes requests for computing resources based on the data received.

The apply module is responsible for passing the new deployment configuration to the Kubernetes API. After that, the next monitoring and scaling cycle is performed.

Results

To evaluate the effectiveness of the proposed method, simulation of the operation of an application that scales both horizontally and vertically is performed. Conducting experiments on a Kubernetes infrastructure is complicated by the fact that in order to update an application's requests, all of its instances need to be restarted. On a real infrastructure, this is achieved by gradually restarting each instance, which does not require significant additional computing resources and does not affect QoS levels. The simulation is done using Python and the Pandas library. The horizontal scaling model is similar to HPA and a reactive approach is used. Target utilisation levels for CPU time and memory are set and requests are reevaluated at 100 second intervals, reflecting the frequency of metric collection and aggregation in a real cluster.

The first experiment allows us to test the work when the priority resource for horizontal scaling is CPU time. Accordingly, memory is scaled vertically. In this study, the total load with a periodicity of 1000 seconds ranges from 200 millicores to 1200 millicores, and the CPU time requirements are set at 250 millicores with a target utilization of 90%. So the application scales from two to five instances. Fig. 5 shows a graph of the workload and the number of instances to process it. Also in Fig. 5, there is ascaling delay when the load appears and, accordingly, the delay before reducing the number of instances, which corresponds to the behaviour of the HPA.

Fig. 6 demonstrates scaling of the memory resource in the first experiment. The initial configuration of memory requests is not optimal and set to 100 megabytes. The model takes into account that each instance has a constant component of memory usage and another one that depends on the load. Accordingly, during the first two periods of operation, memory utilization is 50%. After the component has received a sufficient amount of data, namely 2000 seconds, it scales down memory

ISSN 2788-6603

requests to the target level of 90%. In the fourth period of operation, an atypical peak in memory usage appears and the vertical scaling component responds by increasing memory requests accordingly. After two periods of low utilization, the memory scales down again to 90% utilization during the last two periods of operation.

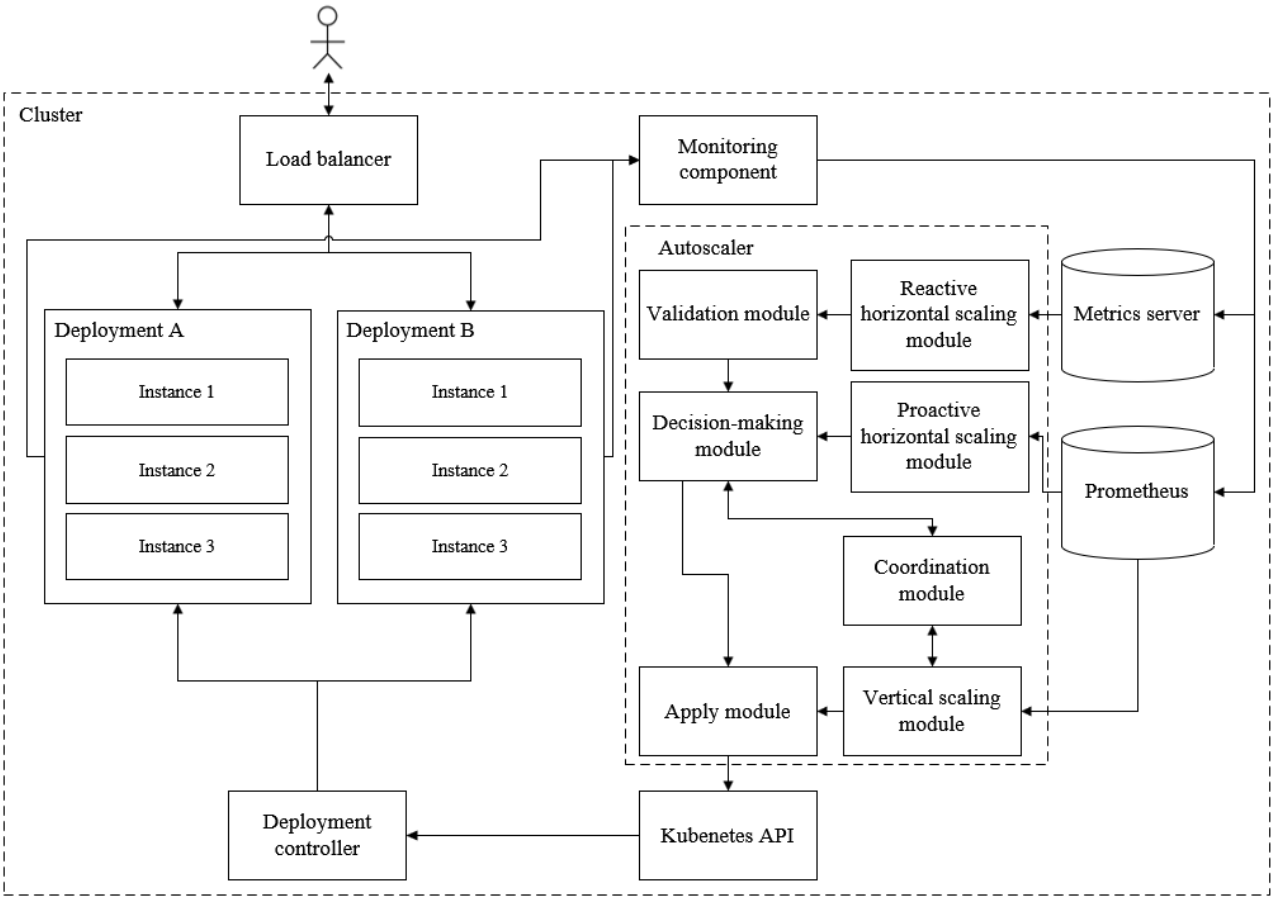


Fig. 4. Structural diagram of a cluster containing hybrid scaling

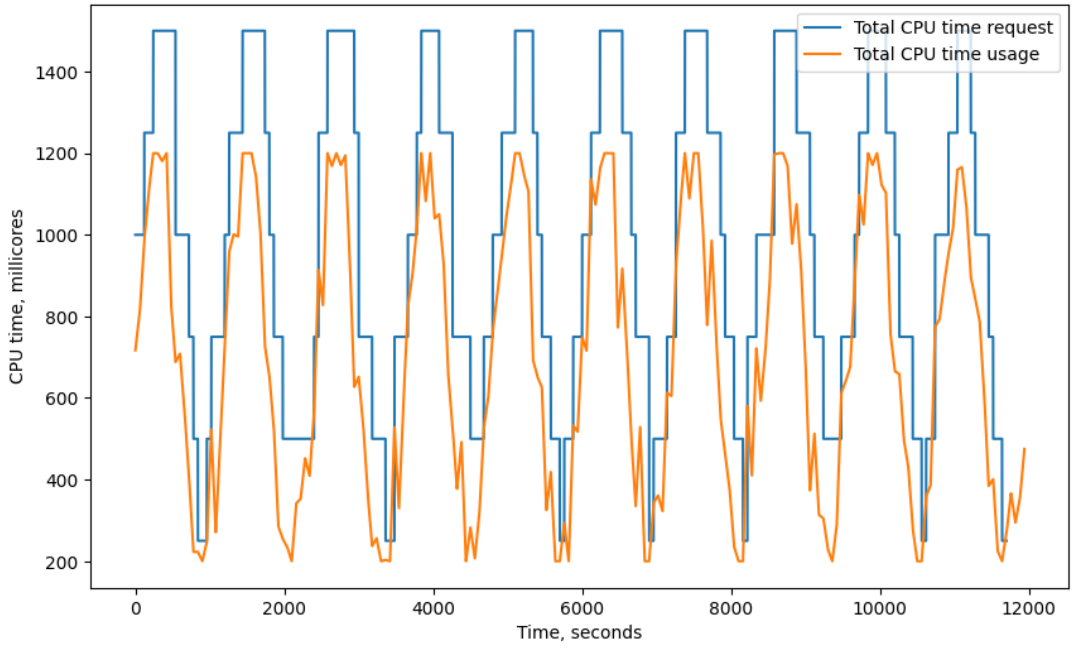


Fig. 5. Horizontal scaling of the application using CPU time

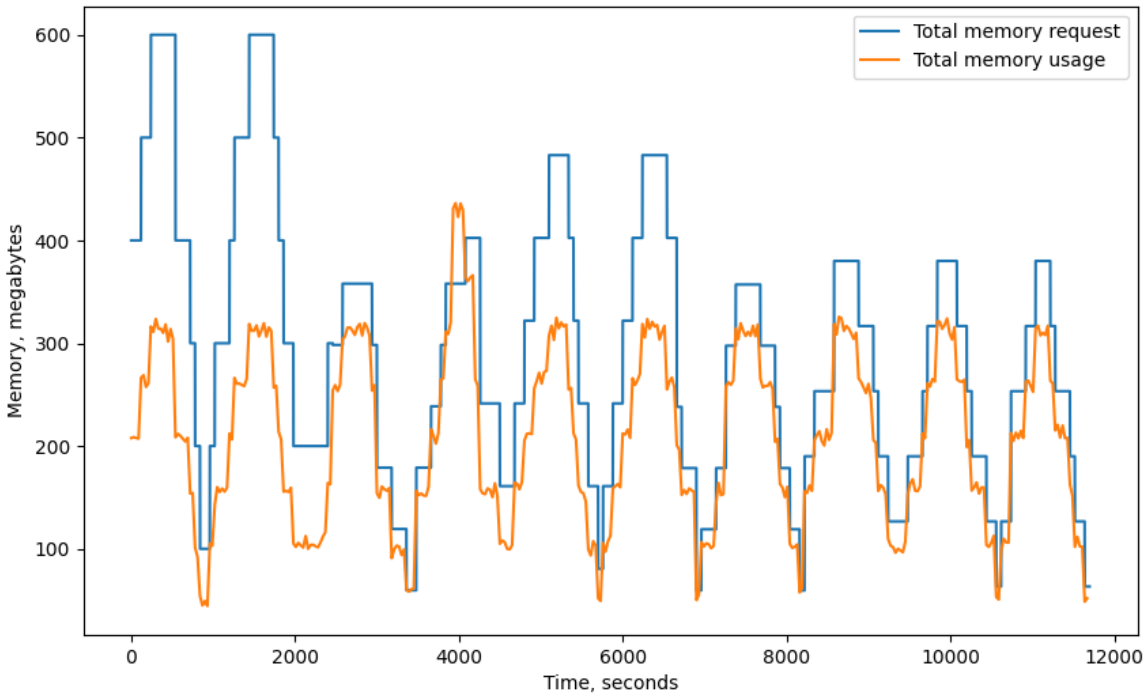


Fig. 6. Vertical scaling of memory requests

The next experiment is intended to evaluate the method's performance when horizontal scaling is performed using memory and CPU time is vertically scaled. Fig. 7 shows the application load and the number of application instances.

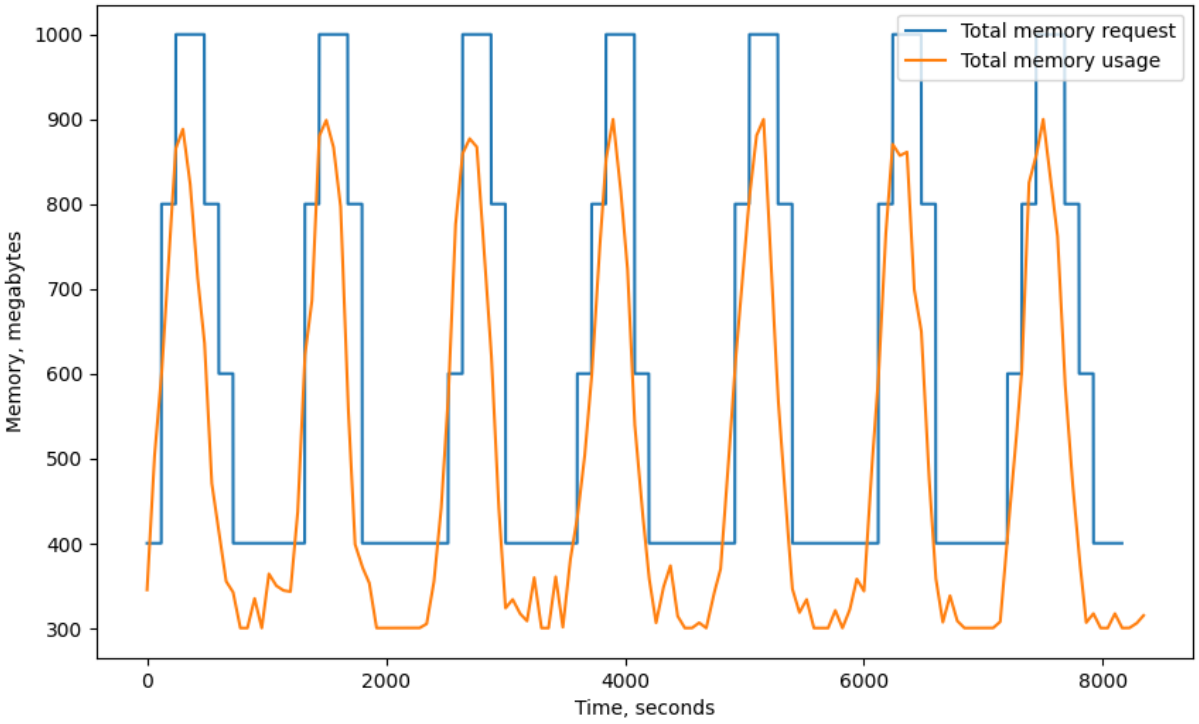


Fig. 7. Horizontal scaling of the application using memory

Fig. 8 shows the results of the vertical scaling. The 95th percentile is used to calculate the CPU time requirements. In Fig. 8, there is a similar pattern of operation for downscaling and upscaling. The difference with the first experiment is the higher level of resource utilization for the vertical scaling.

The last experiment is focused on evaluating the efficiency depending on the length of historical data taken into account during vertical scaling. Horizontal scaling is performed from memory at 1000-second intervals, as shown in Fig. 9.

ISSN 2788-6603

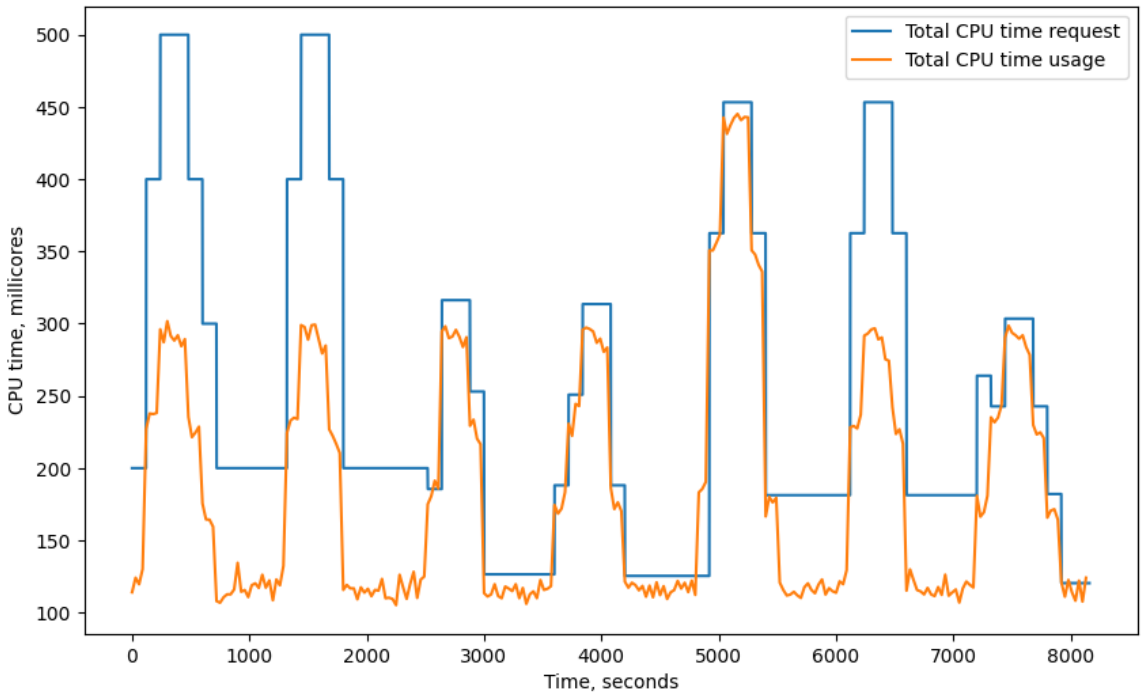


Fig. 8. Vertical scaling of requests using CPU time

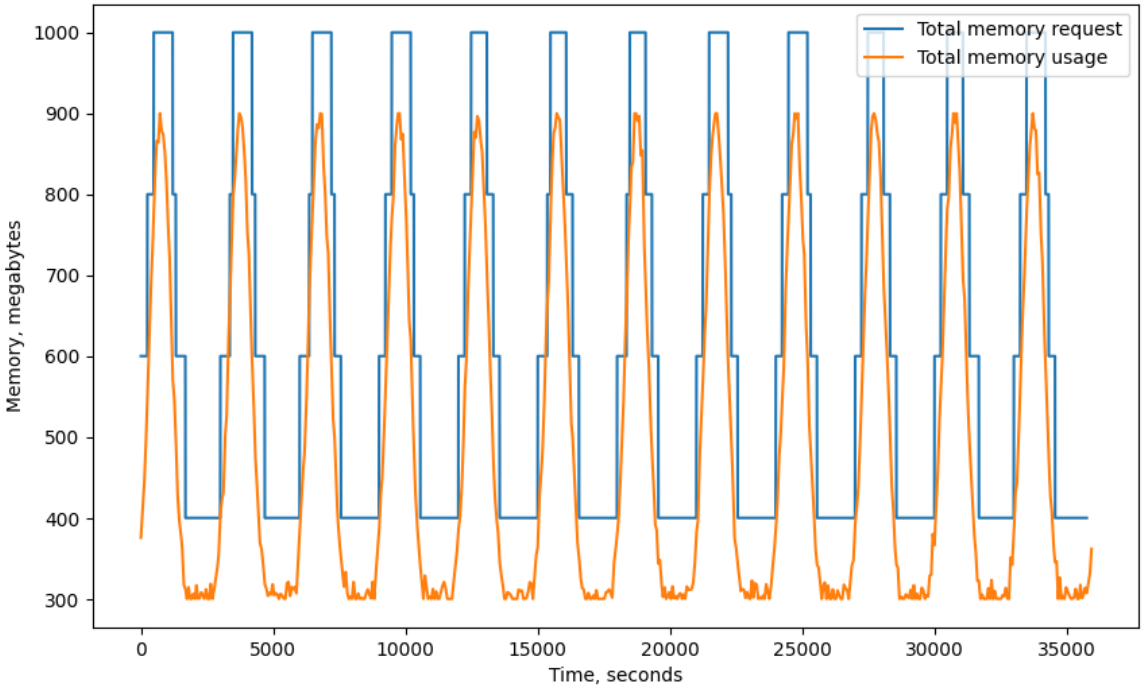


Fig. 9. Horizontal scaling of the application using memory

Fig. 10 shows the results of the vertical scaling component operation. The size of the historical data taken into account is 1000 seconds. The target CPU time utilization is 95%.

A similar experiment is shown in Fig. 11, but the size of the historical data is increased to 4000 seconds:

The results show that scaling occurs with a much shorter period, which, on the one hand, leads to lower resource efficiency and, on the other hand, to better QoS performance at peak loads. This experiment demonstrates the possibility of adapting the developed method to the set SLA requirements.

Further, Fig. 12 compares CPU time utilization for static and dynamic requests. The ratio of resource utilization to requests for dynamic requests ranges from 80% to 120%. In this case, the level of QoS depends on many other factors, including the level of the set limits. At the same time, the use of static requests, at the 95th percentile level, leads to a low utilization rate of 30% to 90% in the given conditions.

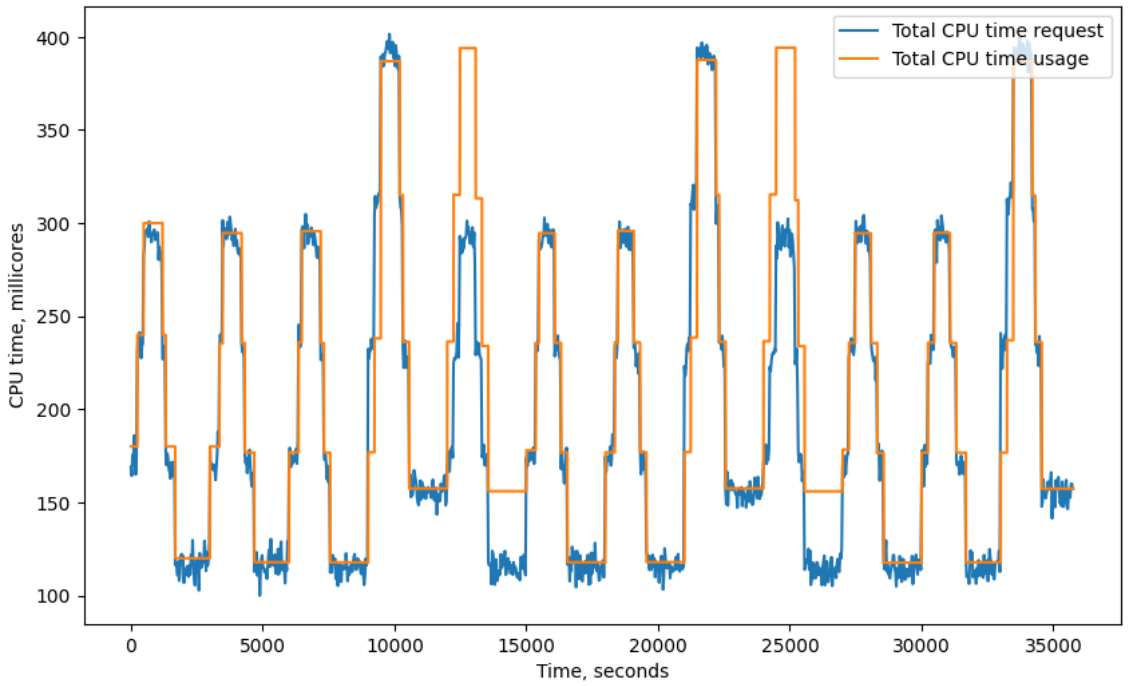


Fig. 10. Vertical scaling of the application for CPU time with a historical data length of 1000 seconds

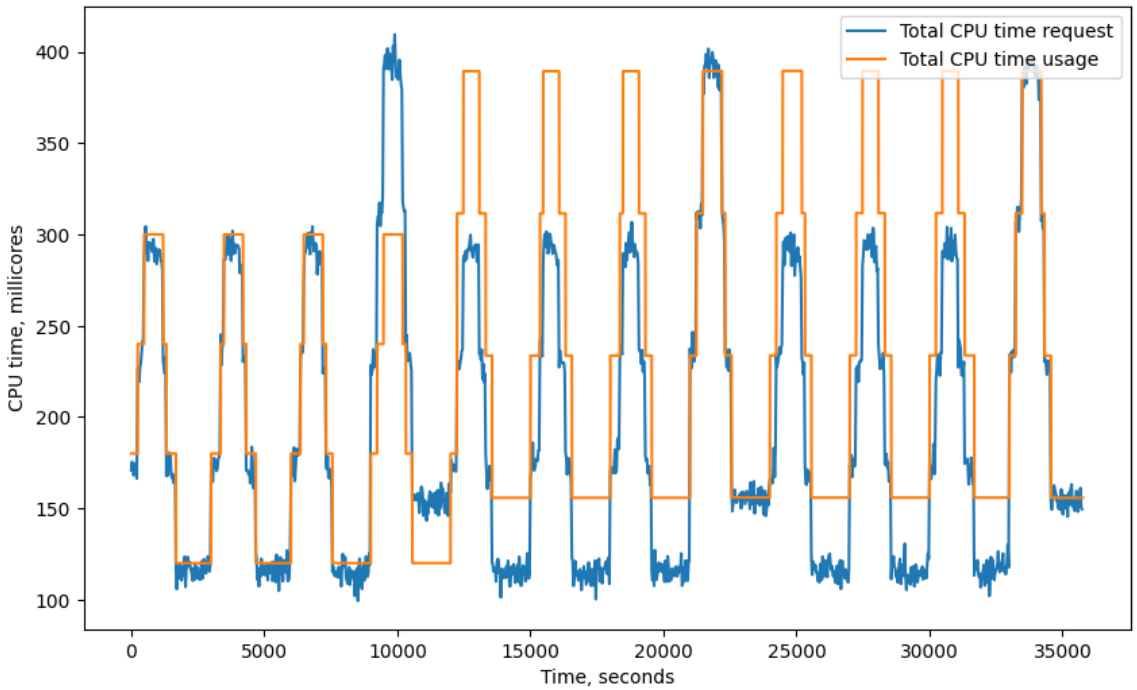


Fig. 11. Vertical scaling of the application by CPU time with 4000 seconds of historical data

The amount of allocated CPU time for dynamic requests is 65% less than for static ones. The conducted experiments are simulations and the results may differ significantly on real infrastructure, but they allow us to evaluate the theoretical possibility of applying the developed method.

Discussion and conclusion

The paper proposes a hybrid scaling method that includes both a horizontal and a vertical component. An optimization method is proposed for resources such as CPU time and network bandwidth. The simulation results show that vertical scaling can increase the efficiency of using cluster computing resources and prove the feasibility of developing a full-featured solution for use in Kubernetes and other orchestration solutions. The developed method proposes reactive management based on current state of the system, but it is possible to use forecasts to manage resource allocation in advance.

ISSN 2788-6603

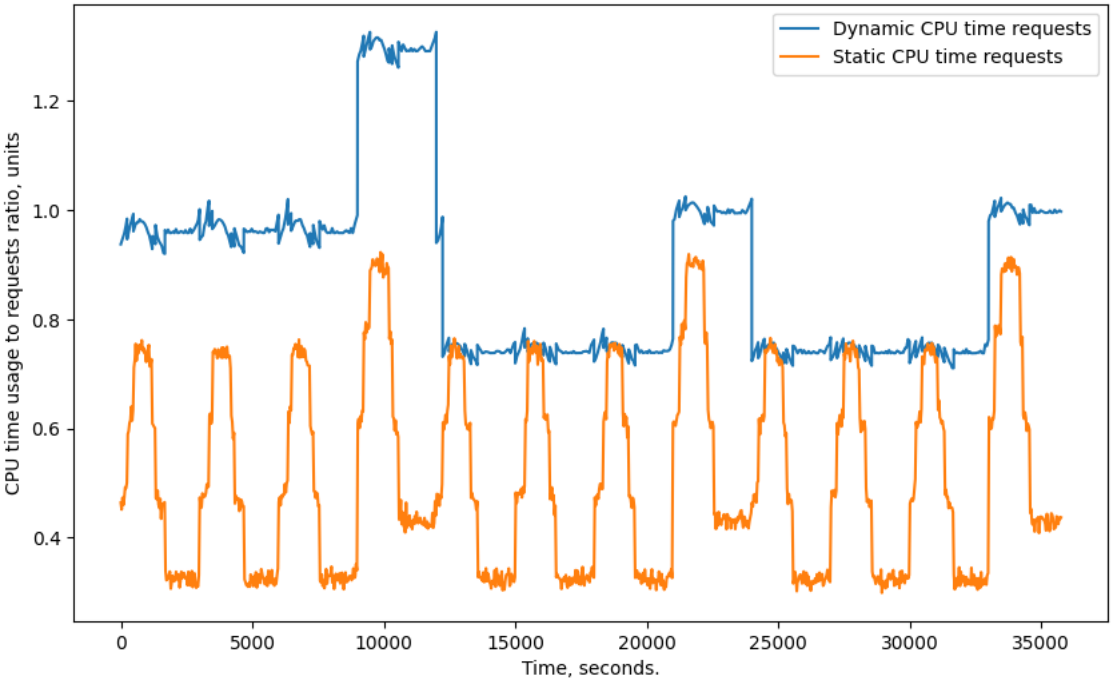


Fig. 12. Comparison of resource utilization with static and dynamic control

Authors' contribution: Vitalii Omelchenko – method development, solution implementation, performing of experiments. Oleksandr Rolik – development of the method and methodology, evaluation of results.

References

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N., & Merle, P. (2017). Elasticity in cloud computing: State of the art and research challenges. *IEEE Transactions on Services Computing*, 11(2), 430–447. Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/TSC.2017.2711009

Al-Haidari, F., Sqalli, M. H., & Salah, K. (2013). Impact of CPU utilization thresholds and scaling size on autoscaling cloud resources. *IEEE 5th International*. In I. Foster, E. Feig, & S. Yau (Eds.) *Conference on Cloud Computing Technology and Science*, 2 (pp. 256–261). Institute of Electrical and Electronics Engineers.

Calheiros, R. N., Toosi, A. N., Vecchiola, C., & Buyya, R. (2012). A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems*, 28(8), 1350–1362. https://doi.org/10.1016/j.future.2012.03.010

Dutta, S., Gera, S., Verma, A., & Viswanathan, B. (2012). SmartScale: Automatic Application Scaling in Enterprise Clouds. In I. Foster, E. Feig, & S. Yau (Eds.). *IEEE 5th International Conference on Cloud Computing* (pp. 221–228). IEEE. https://doi.org/10.1109/cloud.2012.12

Incerto, E., Tribastone, M., & Trubiani, C. (2018). *Combined Vertical and Horizontal Autoscaling Through Model Predictive Control.* In M. Aldinucci, L. Padovani, & M. Torquati (Eds.). *Lecture Notes in Computer Science. Vol. 11014. Parallel Processing* (pp. 147–159). Springer International Publishing. https://doi.org/10.1007/978-3-319-96983-1_11

Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. *Journal of Grid Computing*, 12(4), 559–592. https://doi.org/10.1007/s10723-014-9314-7

Millnert, V., & Eker, J. (2020). HoloScale: Horizontal and vertical scaling of cloud resources. In N. Antonopoulos, O. Rana, & Ch. Jiang (Eds.) *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, 55 (pp. 196–205). IEEE. https://doi.org/10.1109/ucc48980.2020.00038

Omelchenko, V., & Rolik, O. (2022). Automation of resource management in information systems based on reactive vertical scaling. *Adaptive systems of automatic control,* 2(41), 65–78. National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". https://doi.org/10.20535/1560-8956.41.2022.271344

Qu, C., Calheiros, R. N., & Buyya, R. (2018). Auto-scaling web applications in clouds. *ACM Computing Surveys*, 51(4), 1–33. https://doi.org/10.1145/3148149

Quattrocchi, G., Incerto, E., Pinciroli, R., Trubiani, C., & Baresi, L. (2024). Autoscaling Solutions for Cloud Applications Under Dynamic Workloads. In *IEEE Transactions on Services Computing*, 17(3), 804–820. IEEE. https://doi.org/10.1109/tsc.2024.3354062

Rodriguez, M. A., & Buyya, R. (2018). Container-based cluster orchestration systems: A taxonomy and future directions. *Software: Practice and Experiencee,* 49(5), 698–719. https://doi.org/10.1002/spe.2660

Rochman, Y., Levy, H., & Brosh, E. (2014). Efficient resource placement in cloud computing and network applications. *SIGMETRICS Performance Evaluation Review*, 42(2), 49–51. https://doi.org/10.1145/2667522.2667538

Rolik, O., Telenik, S., & Yasochka, M. (2018). *Enterprise IT-infrastructure management.* Naukova dumka [in Ukrainian]. [Ролік О., Теленик С., & Ясочка, М. (2018). *Управління корпоративною IT-інфраструктурою*. Наукова думка].

Rolik, O., & Omelchenko, V. (2024). Proactive horizontal scaling method for Kubernetes. *Radio Electronics, Computer Science, Control*, 1, 221–227. National University "Zaporizhzhia Polytechnic". https://doi.org/10.15588/1607-3274-2024-1-20

Sedaghat, M., Hernandez-Rodriguez, F., & Elmroth, E. (2013). A virtual machine re-packing approach to the horizontal vs. vertical elasticity trade-off for cloud autoscaling. In S. Hariri, & A. Sill (Eds.). *2013 ACM Cloud and Autonomic Computing Conference* (pp. 1–10). ACM. https://doi.org/10.1145/2494621.2494628

Singh, P., Gupta, P., Jyoti, K., & Nayyar, A. (2019). Research on auto-scaling of web applications in cloud: Survey, trends, and future directions. *Scalable Computing: Practice and Experience*, 20(2), 399–432. https://doi.org/10.12694/scpe.v20i2.1537

Straesser, M., Grohmann, J., von Kistowski, J., Eismann, S., Bauer, A., & Kounev, S. (2022). Why is it not solved yet? In D. Feng, & S. Becker (Eds.). *2022 ACM/SPEC International Conference on Performance Engineering* (pp. 105–115). ACM. https://doi.org/10.1145/3489525.3511680

Отримано редакцією журналу / Received: 12.09.24
Прорецензовано / Revised: 23.10.24
Схвалено до друку / Accepted: 07.11.24

Віталій ОМЕЛЬЧЕНКО, асп.
ORCID ID: 0000-0002-3850-6555
e-mail: vitaly.om25@gmail.com
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

Олександр РОЛІК, д-р техн. наук, проф.
ORCID ID: 0000-0001-8829-4645
e-mail: arolick@gmail.com
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

ГІБРИДНИЙ МЕТОД ГОРИЗОНТАЛЬНОГО І ВЕРТИКАЛЬНОГО МАСШТАБУВАННЯ ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ

В с т у п . ***Підвищення ефективності використання обчислювальних ресурсів кластера при дотриманні встановлених рівнів QoS є критично важливим завданням у керуванні IT-інфраструктурою. Динамічне керування обчислювальними ресурсами, зокрема вертикальне та горизонтальне масштабування, є інструментом, що дозволяє автоматизувати процеси адаптації застосунків до динамічних навантажень. Метою цієї роботи є підвищення ефективності існуючих методів масштабування за допомогою їхнього комбінування.***

М е т о д и . ***Запропоновано гібридний метод масштабування з використанням модуля координації. Цей модуль узгоджує роботу компонентів вертикального та горизонтального масштабування на основі заданих обмежень, пріоритетів і поточного стану системи. Модуль координації відповідає за підвищення ефективності обох компонентів масштабування і запобігає неузгодженості у процесі конфігурації кількості екземплярів застосунку та запитів на обчислювальні ресурси.***

Р е з у л ь т а т и . ***Для досягнення поставленої мети розроблено гібридний метод вертикального та горизонтального масштабування з використанням координації компонентів на основі пріоритетів. Пріоритетна конфігурація визначає порядок роботи компонентів під час узгодження їхньої роботи. У випадку горизонтально-вертикального порядку непріоритетний вертикальний компонент не впливає на конфігурацію пріоритетного горизонтального компонента. Проведено оцінювання розробленого методу на основі моделювання роботи застосунку з навантаженням, що містить постійну сезонність. Експерименти демонструють зменшення збиткового резервування обчислювальних ресурсів кластера на 65 % порівняно зі статичними запитами.***

В и с н о в к и . ***Розроблений метод можна застосувати для підвищення ефективності використання ресурсів у кластерах під час динамічних навантажень порівняно з базовими методами масштабування. У подальших дослідженнях необхідно оцінити метод на реальній інфраструктурі. Також варто дослідити роботу гібридного методу з використанням предиктивного підходу.***

К л ю ч о в і с л о в а : ***інформаційні технології, інформаційні системи, керування ресурсами, масштабування, IT-інфраструктура, вертикальне масштабування, горизонтальне масштабування.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

УДК 681.324
DOI: https://doi.org/10.17721/AIT.2024.1.06

Олег БАРАБАШ, д-р техн. наук, проф.
ORCID ID: 0000-0003-1715-0761
e-mail: bar64@ukr.net
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

Андрій МАКАРЧУК, асп.
ORCID ID: 0000-0002-6422-7488
e-mail: andreymakarh2@gmail.com
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

РОЗРОБЛЕННЯ НОВОГО ПОКАЗНИКА ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ ТА ЙОГО ОЦІНЮВАННЯ ЗА ДОПОМОГОЮ БАГАТОВИМІРНОЇ ПОЛІНОМІАЛЬНОЇ РЕГРЕСІЇ

В с т у п . ***Функціональна стійкість розподілених систем стає все важливішим поняттям із розвитком інформаційних технологій. Набуває актуальності певна формалізація цього поняття. Математична формалізація функціональної стійкості у вигляді її показників і критеріїв відбувається вже не перше десятиліття. Особливу роль тут відіграють саме показники функціональної стійкості, яких сформульовано вже не так і мало. Однак головним недоліком більшості з них є те, що окрім своєї високої обчислювальної складності, ці показники ще й залежать від багатьох інших параметрів, або те, що вони недостатньо повно описують функціональну стійкість розглядуваної розподіленої системи. В цій роботі введено новий показник функціональної стійкості, у якого відсутній другий із згаданих недоліків. Перший недолік усувають завдяки використанню оцінювання, основаного на багатовимірній поліноміальній регресії.***

М е т о д и . ***Використано методи комп'ютерного моделювання та наближення.***

Р е з у л ь т а т и . ***Як спосіб розроблення нового показника функціональної стійкості вибрано модифікацію існуючого показника, відомого в літературі як імовірність справності. Висунувши певні припущення та провівши деякі перетворення, отримано величину, яка має певні хороші властивості, а саме: вказана величина міститься строго в проміжку від нуля до одиниці, а також чим більшою вона є, тим більш функціонально стійкою можна вважати розглядувану розподілену систему. Проте отриманий показник функціональної стійкості вимагає дуже багато обчислень, тому здійснено спробу побудови оцінки цього показника за допомогою методів наближення. В межах указаного дослідження було вивчено можливості застосування багатовимірної поліноміальної регресії. Як показало комп'ютерне моделювання, для досягнення точності, рівної, в середньому, двом відсоткам, достатньо використовувати п'ятивимірну поліноміальну регресію четвертого степеня. Подальше підвищення степеня п'ятивимірної регресійної моделі не даватиме значущого зменшення похибки.***

В и с н о в к и . ***Введений у роботі показник функціональної стійкості є зручним засобом для дослідження функціональної стійкості розподілених систем. Однак він вимагає значної кількості обчислень. Із цією метою представлено метод оцінювання вказаного показника, який дозволяє достатньо точно обчислити введений показник функціональної стійкості.***

К л ю ч о в і с л о в а : ***функціональна стійкість, показник, наближення, оптимізація, регресія, функції багатьох змінних.***

Вступ

Проєктування й експлуатація розподілених систем стає все актуальнішим напрямом через збільшення спектра задач, зокрема і пов'язаних із телекомунікаціями, обміном та аналізом даних, способами організації складних обчислень тощо. На основі цього очікуваним є виникнення поняття функціональної стійкості, тобто можливості бодай часткового виконання розподіленою системою поставлених перед нею задач з урахуванням сторонніх ефектів, та методів формального її оцінювання.

Розроблені на сьогодні показники функціональної стійкості, як-от імовірність і матриця зв'язності, степінь вершинної та реберної зв'язності тощо, мають бодай один із таких недоліків:

- висока обчислювальна складність;
- складна інтерпретовуваність;
- значна залежність від досить великої кількості параметрів;
- відсутність ефективних методів оцінювання.

Одним із можливих способів розв'язання окресленої проблеми є розроблення нових показників функціональної стійкості та відповідних методів їх оцінювання. В цій роботі запропоновано новий структурний показник функціональної стійкості. Попри його специфіку, цей показник ефективно можна оцінювати за допомогою багатовимірної поліноміальної регресії, що буде продемонстровано в основній частині роботи.

Огляд останніх публікацій. Широке застосування розподілених систем нині вимагає більше уваги приділяти питанням, пов'язаним із функціональною стійкістю. Зокрема це стосується і формальних методів опису функціональної стійкості розподілених систем. Вказаною проблемою займається чимало науковців. Наприклад, у роботах (Барабаш, 2004; Барабаш, & Кравченко, 2002) виконано огляд чималої кількості вже розроблених показників і критеріїв функціональної стійкості, а також умов, виконання яких функціональну стійкість забезпечує. В дослідженні (Миронюк та ін., 2024) розглянуто деякі з тих самих показників, однак, на фоні вже сучасного стану інформаційних технологій. Певні показники функціональної стійкості досліджено більш детально. Наприклад, в роботі (Барабаш та ін., 2024) досліджено застосування відомих методів наближення (Shidlich, 2013), (Abdullayev et al., 2019) для оцінювання ймовірності зв'язності. В ряді робіт, як-от: (Kravchenko, Leschenko, & Mykus, 2016), (Mashkov et al. 2021), (Kovalchuk et al., 2020), демонструються певні можливості застосування поняття та показників функціональної стійкості в конкретних ситуаціях. Однак, такі роботи часто не враховують саме специфіку показників функціональної стійкості, акцентуючи, зазвичай, увагу лише на можливостях прикладного застосування.

Багато нині розроблених показників функціональної стійкості мають, як мінімум, одну з двох проблем: конкретний показник може залежати від дуже великої кількості параметрів, що робить дослідження функціональної стійкості

досить проблематичним, або його використання вимагає дуже великої кількості обчислень. Спроби розв'язання другої проблеми здійснювалися в деяких роботах типу (Барабаш та ін., 2024), але такі спроби, переважно, мають недостатньо глибокий характер. Спроб розв'язання першої проблеми майже не було. Саме тому постає необхідність в розробленні нових показників функціональної стійкості, простіших у плані прикладного застосування, та, у випадку їхньої високої обчислювальної складності, методів ефективного їх оцінювання.

Опис нового показника функціональної стійкості. Розглянемо інформаційну систему, структуру якої можна представити неорієнтованим графом $G=G(V,L)$. Під імовірністю зв'язності R_{ij} розуміють імовірність передачі інформації між машинами v_i та v_j. Оскільки ймовірність зв'язності R_{ij} математично можна розглядати як функцію не лише від графа G, а й від імовірності справності кожного з елементів розглядуваної системи, тобто $\forall i,j=1,2,...,n: R_{ij}=R_{ij}\left(G,p(v_1),p(v_2),...,p(v_n),p(l_1),p(l_2),...,p(l_m)\right)$, то логічним є питання, як перетворити R_{ij} так, щоб залежність, у результаті, спостерігалася лише від графа G? Іншими словами, яке перетворення для R_{ij} необхідно підібрати, щоб воно залежало лише від структури розглядуваної інформаційної системи?

Вказану проблему можна розв'язати у такий спосіб. Для початку покладемо, що на момент обчислення показника функціональної стійкості всі машини в системі є абсолютно надійними, тобто $\forall i=1,2,...,n: p(v_i)=1$. Тепер вважатимемо, що ймовірності справності всіх ліній зв'язку однакові та рівні числу p, тобто $\forall j=1,2,...,m: p(l_j)=p$. З урахуванням описаних припущень неважко побачити, що тоді ймовірність зв'язності R_{ij} буде функцією від графа G та величини p, тобто $R_{ij}=R_{ij}(G,p)$. Введемо тепер ще позначення:

$$\bar{R}_{ij}=\bar{R}_{ij}(G)=\int_0^1 R_{ij}(G,p)\,dp.$$

На основі цієї формули можна ввести новий структурний показник функціональної стійкості, який позначимо $\bar{R}$ та обчислюватимемо за формулою

$$\bar{R}=\min_{i,j=1,2,...,n\wedge i\neq j}\bar{R}_{ij}. \quad (1)$$

Розглянемо певні особливості показника функціональної стійкості (1). По-перше, величина (1) буде дійсною величиною, яка коливатиметься в проміжку від 0 до 1. По-друге, збільшення цієї величини явно вказуватиме на "покращення" функціональної стійкості. Іншими словами, чим ближчий показник (1) до одиниці, тим більш функціонально стійкою буде розглядувана розподілена система. Також не варто забувати про те, що (1), фактично, залежить виключно від структури розглядуваної системи, що, у свою чергу, дозволяє певною мірою абстрагуватися від таких параметрів елементів системи, як наприклад, імовірність справності. Однак, попри перераховані вище переваги, показник (1) має один дуже суттєвий недолік: складність його обчислення становить $\Omega\left(n^2 2^n\right)$ (Барабаш, 2004; Cormen et al., 2009), де n – кількість машин у системі, що може бути неприпустимим з урахуванням сучасних габаритів розподілених інформаційних систем. Тому логічним є питання про побудову методів оцінювання показника (1), які вимагали б значно менше обчислень. Іншими словами, є необхідність в оптимізації обчислення показника (1).

Методи

З огляду на поставлену задачу та характер проблеми, в роботі запропоновано використовувати методи комп'ютерного моделювання та наближення.

Оцінювання значення показника функціональної стійкості (1) за допомогою поліноміальної регресії. Відомо, що одним із можливих підходів до оптимізації обчислень є використання методів наближення. Оскільки методів наближення зараз розроблено досить багато, то виникає необхідність у виборі підходящого. Для початку покладемо, що показник $\bar{R}$ є функцією від певних числових характеристик графа G, який описує структуру розглядуваної розподіленої системи. З урахуванням того, що показник (1) як функція нам не заданий (його значення ми, фактично, можемо обчислити лише для конкретної структури розподіленої системи), одним із найпростіших підходів є використання степеневих поліномів, побудованих за допомогою методу найменших квадратів, тобто застосування багатовимірної поліноміальної регресії.

Тепер визначимося, які числові параметри графа G досліджуватимемо як аргументи розглядуваної функції. Мабуть, інтуїтивно найзрозумілішими такими характеристиками будуть мінімальний і максимальний степені вершини у графі G, які позначимо як $d_{\min}$ та $d_{\max}$, відповідно, та насиченість графа S. Однак, оскільки перші два параметри, фактично, є натуральними числами, то для зручності подальшого дослідження введемо величини $\eta_{\min}=\frac{d_{\min}}{n-1}$ та $\eta_{\max}=\frac{d_{\max}}{n-1}$, де n –кількість машин у розглядуваній інформаційній системі. Отже, покладемо, що показник $\bar{R}$ є функцією від параметрів $\eta_{\min}$, $\eta_{\max}$ та S, тобто $\bar{R}=\bar{R}(\eta_{\min},\eta_{\max},S)$.

Як зазначено вище, розглядатимемо поліноміальну регресійну модель (надалі іноді називатимемо довільну поліноміальну регресійну модель степеневим поліномом або просто поліномом), яку позначимо як $P_v=P_v(\eta_{\min},\eta_{\max},S)$, де v – степінь полінома P_v. Для її побудови та подальшого дослідження згенеровано вибірку 9546 різних структур

ISSN 2788-6603

інформаційних систем невеликих розмірів із відповідними значеннями параметрів $\eta_{\min}$, $\eta_{\max}$ і S та обчисленого показника (1) за допомогою методу Монте-Карло.

Неважко побачити, що в межах поставленої задачі розглядати лінійну регресійну модель, тобто P_1, навряд чи доречно. Тому почнемо дослідження з випадку так званої квадратичної регресії, тобто коли $v = 2$. В цьому випадку степеневий поліном P_v матиме вигляд

$$P_v = P_2(\eta_{\min}, \eta_{\max}, S) = \\ = a_2\eta_{\min}^2 + a_1\eta_{\min} + b_2\eta_{\max}^2 + b_1\eta_{\max} + c_2S^2 + c_1S + d_1\eta_{\min}\eta_{\max} + d_2\eta_{\min}S + d_3\eta_{\max}S + e. \quad (2)$$

Якщо розв'язати задачу мінімізації середньоквадратичного відхилення методом градієнтного спуску, то можна побачити, що квадратична регресійна модель $P_2 = P_2(\eta_{\min}, \eta_{\max}, S)$ запишеться як

$$P_2(\eta_{\min}, \eta_{\max}, S) = \\ = -4.28\eta_{\min}^2 + 0.95\eta_{\min} - 0.35\eta_{\max}^2 + 0.25\eta_{\max} - 3.74S^2 + 2.68S - 0.45\eta_{\min}\eta_{\max} + 4.5\eta_{\min}S + 0.74\eta_{\max}S. \quad (3)$$

Тепер оцінимо середню абсолютну й відносну похибки під час використання степеневого полінома (3) для оцінювання показника функціональної стійкості (1), базуючись на згенерованій вибірці.

Результати

Провівши необхідні обчислення на наявних даних, можна встановити, що середня абсолютна похибка за використання (3) приблизно рівна 0,038, а середня відносна – 6,2 %. На основі цього результату може виникнути питання, чи можливо зменшити середню абсолютну та середню відносну похибку, розглядаючи випадок $v > 2$? Провівши аналогічні обчислення, але для степеневих поліномів P_v вище другого степеня, можна отримати результати, представлені в табл. 1.

Таблиця 1

Точність поліноміальної регресійної моделі P_v як методу оцінювання показника функціональної стійкості $\bar{R}$

P_v	Середня абсолютна похибка	Середня відносна похибка, %
P_3	0,03348	5,64
P_4	0,03233	5,52
P_5	0,04421	7,19
P_6	0,03193	5,44
P_7	0,03069	5,26

Як видно з табл. 1, маємо чітку відповідь на поставлене питання: випадок $v > 2$ не дає значущого покращення в точності (що, зокрема і видно по середній відносній похибці), особливо, на фоні підвищення складності обчислень. Це означає, якщо в межах поставленої задачі допустимо мати точність оцінювання показника функціональної стійкості $\bar{R}$, в середньому ± 6 %, то використання полінома (3) буде цілком достатньо. Однак, чи можливо покращити точність оцінювання цього показника функціональної стійкості за допомогою поліноміальної регресії, не обмежуючись лише підвищенням її степеня?

Щоб відповісти на це питання, розглянемо показник $\bar{R}$ як функцію не від трьох, а від п'яти числових показників структури інформаційної системи. Для цього розглянемо ще два числові параметри графа G, що описує структуру розглядуваної інформаційної системи, а саме степінь вершинної зв'язності графа κ_G та степінь реберної зв'язності графа λ_G. Відтак, діючи так само, як і з мінімальним та максимальним степенем вершини графа G, можна розглянути дві додаткові змінні, а саме $q_G = \frac{\kappa_G}{n-1}$ та $u_G = \frac{\lambda_G}{n-1}$. Далі розглянемо показник (1) як функцію від $\eta_{\min}$, $\eta_{\max}$, S, q_G та u_G, тобто тепер вважатимемо, що $\bar{R} = \bar{R}(\eta_{\min}, \eta_{\max}, S, q_G, u_G)$, і, діючи аналогічно, наблизимо його за допомогою поліноміальної регресійної моделі $\tilde{P}_v = \tilde{P}_v(\eta_{\min}, \eta_{\max}, S, q_G, u_G)$, де знову v – степінь полінома P_v.

Діючи аналогічно, розглянемо спочатку квадратичну регресійну модель, тобто випадок $v = 2$. У цьому випадку розглядуваний степеневий поліном $\tilde{P}_v$ матиме вигляд

$$\tilde{P}_v = \tilde{P}_2(\eta_{\min}, \eta_{\max}, S, q_G, u_G) = \\ = -5.169\eta_{\min}^2 + 2.74\eta_{\min} + 0.2754\eta_{\max}^2 + 0.885\eta_{\max} - 2.07S^2 + 1.476S + 4.599q_G^2 - 3.772q_G - \\ -0.357u_G^2 - 1.779u_G - 0.748\eta_{\min}\eta_{\max} + 2.607\eta_{\min}S - 1.858\eta_{\min}q_G + 2.175\eta_{\min}u_G - \\ -0.858\eta_{\max}S - 3.248\eta_{\max}q_G + 0.808\eta_{\max}u_G + 4.963Sq_G + 0.245Su_G + 2.07q_Gu_G. \quad (4)$$

Провівши необхідні обчислення, можемо побачити, що середня абсолютна похибка за використання полінома (4) для оцінювання показника $\overline{R}$ становить приблизно 0.025, а середня відносна – приблизно 3,9 %. Іншими словами степеневий поліном (4) дозволяє точніше оцінити введений вище показник функціональної стійкості (1), аніж степеневий поліном (3). Очевидним постає питання, наскільки зміниться точність оцінювання показника $\overline{R}$ у разі збільшення степеня регресійної моделі $\tilde{P}_v$? Для цього розглянемо табл. 2.

Таблиця 2

Точність регресійної моделі (4) за підвищення степеня

$\tilde{P}_v$	**Середня абсолютна похибка**	**Середня відносна похибка, %**
$\tilde{P}_3$	0,013	2,27
$\tilde{P}_4$	0,009	1,76
$\tilde{P}_5$	0,008	1,62
$\tilde{P}_6$	0,009	1,7
$\tilde{P}_7$	0,007	1,53

Із табл. 2 чітко видно, що, на відміну від поліноміальної регресійної моделі $P_v = P_v\left(\eta_{\min}, \eta_{\max}, S\right)$, поліноміальна регресійна модель $\tilde{P}_v = \tilde{P}_v\left(\eta_{\min}, \eta_{\max}, S, q_G, u_G\right)$ даватиме значно кращу точність оцінювання показника функціональної стійкості (1). Щоб цей факт відслідкувати наглядніше, побудуємо графік, на якому по осі *x* відкладемо степінь поліноміальної регресії v, а по осі *y* – значення похибки.

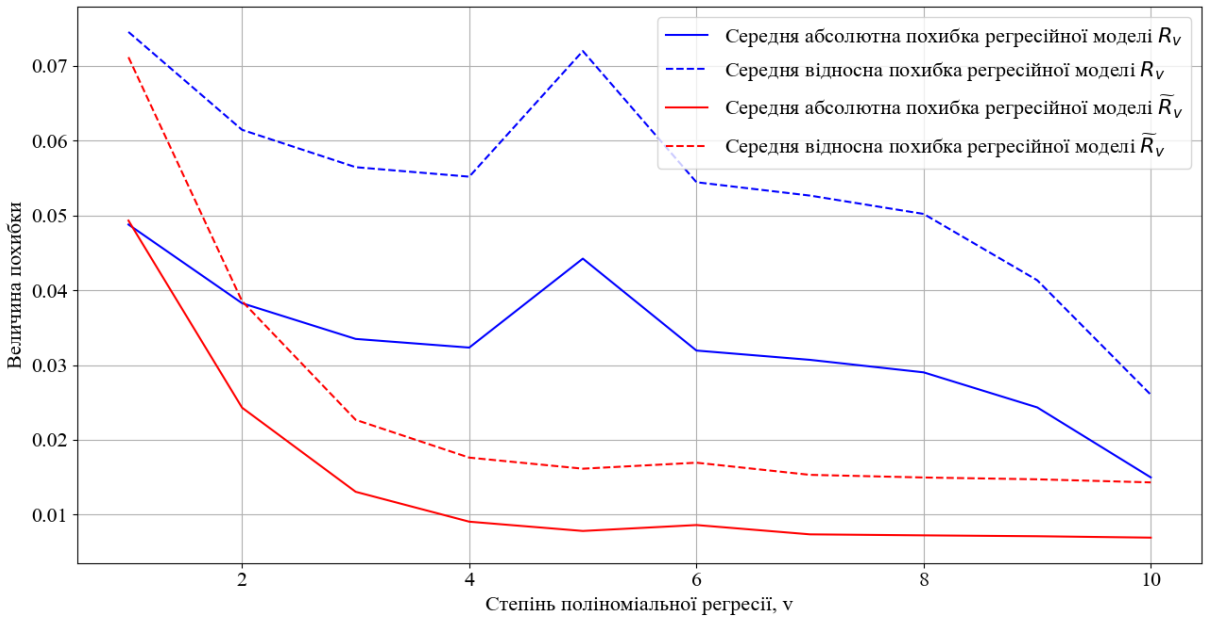


Рис. 1. Порівняння поведінки середньої абсолютної та середньої відносної похибки оцінювання показника (1) поліноміальними регресійними моделями P_v та $\tilde{P}_v$ залежно від степеня

Як видно з рис. 1, середня абсолютна та середня відносна похибка, за якою можна оцінити показник (1) за допомогою згаданих регресійних моделей, поводитимуться подібно. Однак, можна також помітити, якщо оцінювати показник (1) за допомогою моделі $\tilde{P}_v$, а не P_v, тобто, спираючись на 5 зазначених вище числових характеристик структури інформаційної системи, а не на 3, то середня абсолютна та середня відносна похибки такого оцінювання будуть більш прогнозованими, з одного боку, а якість такого оцінювання буде навіть краща. Помічаємо, що за оцінювання показника (1) за допомогою поліноміальної регресійної моделі $\tilde{P}_v$, розгляд випадку $v > 4$ не має особливого сенсу, оскільки значущого покращення точності оцінювання на фоні підвищення кількості обчислень не спостерігатиметься (рис. 2).

Зважаючи на викладене, логічним було б вважати, що оптимальним способом оцінювання показника (1) при використанні багатовимірної поліноміальної регресії буде застосування поліномів $\tilde{P}_3$ та $\tilde{P}_4$: перший вимагає менше

ISSN 2788-6603

обчислень та легший у використанні, а другий дозволяє робити точніше оцінювання. Проте з іншого боку, для обчислених значеннях параметрів $\eta_{\min}$, $\eta_{\max}$, S, q_G та u_G обидва поліноми мають однакову обчислювальну складність незалежно від габаритів розглядуваної розподіленої системи (рис. 3 та 4).

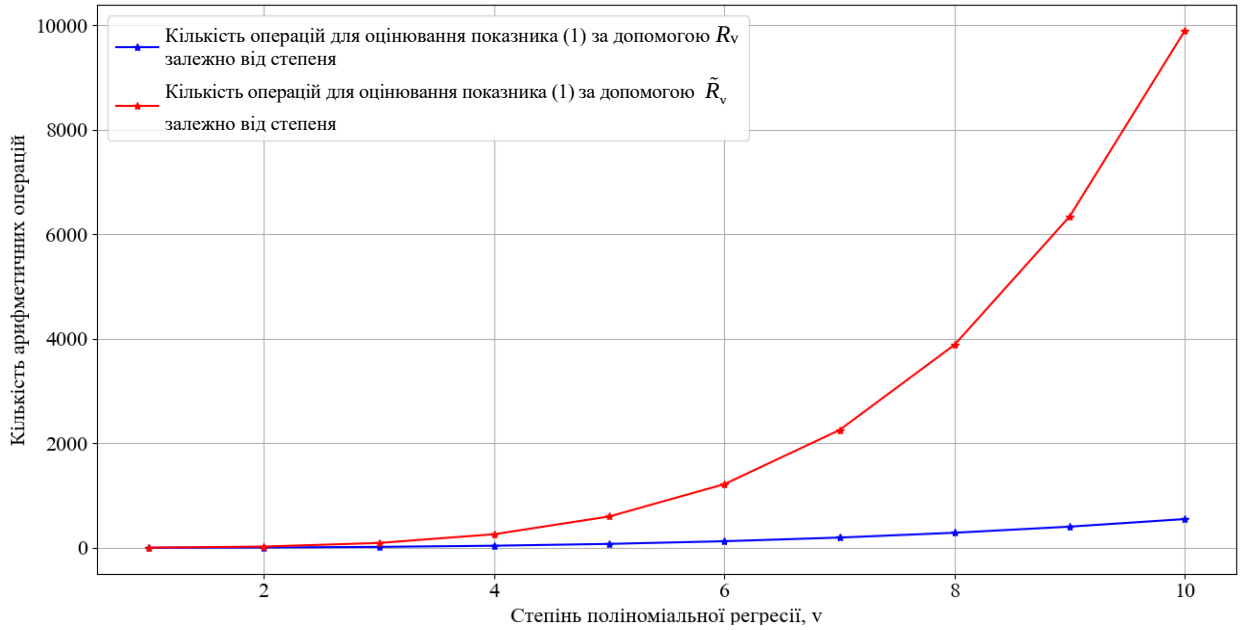


Рис. 2. Порівняння кількості обчислень, необхідних для оцінювання показника функціональної стійкості (1) за допомогою регресійних моделей $\tilde{P}_v$ та P_v залежно від степеня

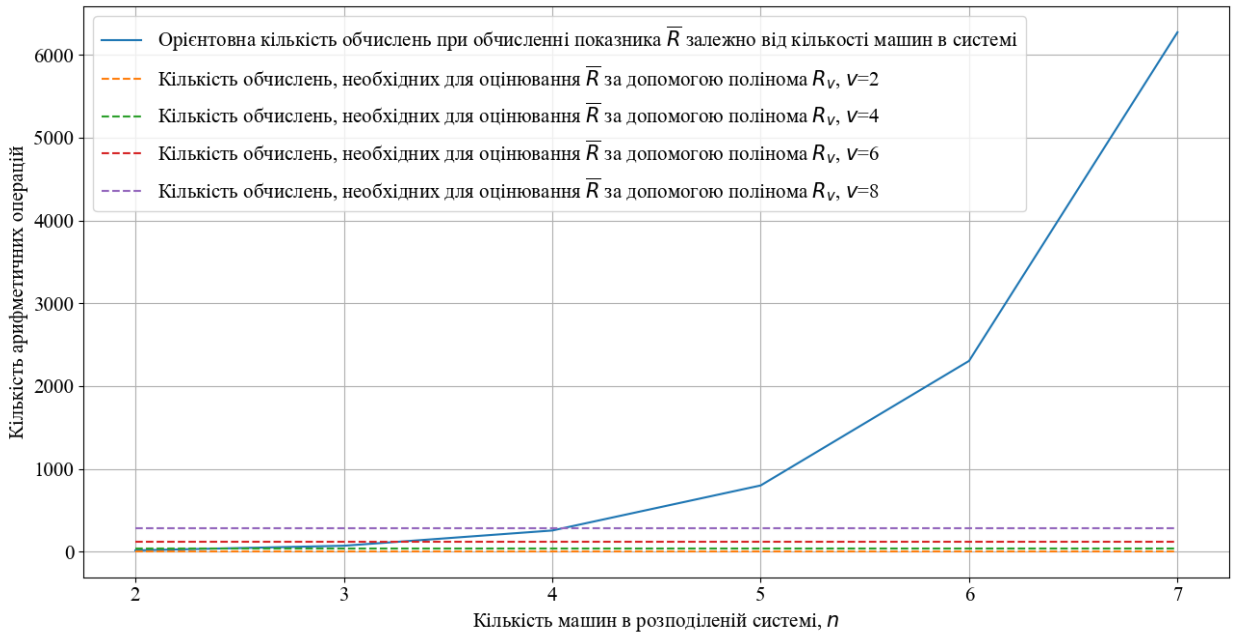


Рис. 3. Порівняння необхідної кількості обчислень під час безпосереднього обчислення показника функціональної стійкості (1) та за його оцінювання за допомогою полінома P_v залежно від кількості машин у системі

На рис. 3 та 4 видно, що габарити розглядуваної розподіленої системи справді безпосередньо не впливають на кількість обчислень у разі використання поліномів $\tilde{P}_v$ та P_v для оцінювання показника функціональної стійкості (1).Це означає, що застосування регресійних моделей для оцінювання введеного в роботі показника функціональної стійкості (1) дійсно може бути робочим рішенням. Однак виникає питання: поліноміальну регресію якого степеня доречно брати для оцінювання показника (1)?

З огляду на табл. 1 та рис. 1, видно, що тривимірна поліноміальна регресія, тобто поліноми P_v можуть забезпечувати необхідну точність обчислень лише коли степінь регресійного полінома v є досить великим. Це, у свою чергу може вимагати занадто багато обчислень. З урахуванням цього подальший розгляд тривимірної поліноміальної регресії як методу оцінювання введеного раніше показника функціональної стійкості можна вважати недоцільним. Тому дослідимо в цьому ракурсі лише п'ятивимірну поліноміальну регресію, тобто поліноми $\tilde{P}_v$.

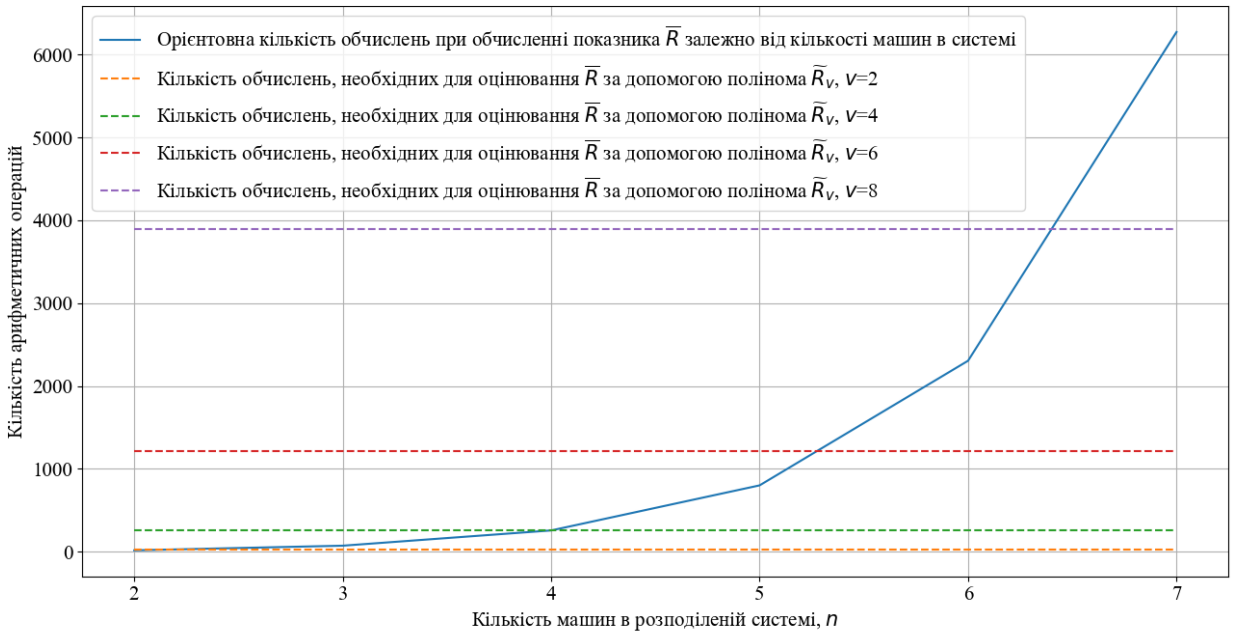


Рис. 4. Порівняння необхідної кількості обчислень під час безпосереднього обчислення показника функціональної стійкості (1) та за його оцінювання за допомогою полінома $\tilde{P}_v$ залежно від кількості машин у системі

Оскільки аналіз лінійної та квадратичної п'ятивимірної регресії як методу оцінювання показника (1) уже здійснено, то розглянемо кубічний випадок, тобто поліном $\tilde{P}_3$. Якщо його обчислювати так, як описано вище в роботі, використовуючи при цьому ту саму вибірку, то вказаний поліном запишеться так:

$$\tilde{P}_3 = \tilde{P}_3(\eta_{\min}, \eta_{\max}, S, q_G, u_G) =$$
$$= 7.01\eta_{\min} + 2.57\eta_{\max} + 1.14S - 13.69q_G - 3.84u_G - 14\eta_{\min}^2 - 4.56\eta_{\min}\eta_{\max} + 5.6\eta_{\min}S - 21.83\eta_{\min}q_G + 7.54\eta_{\min}u_G -$$
$$-1.63\eta_{\max}^2 - 2.11\eta_{\max}S - 6.53\eta_{\max}q_G + 7.05\eta_{\max}u_G - 7.96S^2 + 32.52Sq_G + 0.19Su_G + 35.09q_G^2 + 4.38q_Gu_G - 2.89u_G^2 +$$
$$+12.98\eta_{\min}^3 - 1.66\eta_{\min}^2\eta_{\max} - 7.23\eta_{\min}^2S + +12.31\eta_{\min}^2q_G + 4.42\eta_{\min}^2u_G + 0.72\eta_{\min}\eta_{\max}^2 + 7.89\eta_{\min}\eta_{\max}S + 4.66\eta_{\min}\eta_{\max}q_G -$$
$$-1.74\eta_{\min}\eta_{\max}u_G - 7.23\eta_{\min}S^2 + 17.93\eta_{\min}Sq_G - 13.87\eta_{\min}Su_G + 6.99\eta_{\min}q_G^2 - 4.48\eta_{\min}q_Gu_G - 0.28\eta_{\min}u_G^2 - 0.49\eta_{\max}^3 +$$
$$+5.21\eta_{\max}^2S + 3.09\eta_{\max}^2q_G - 4.03\eta_{\max}^2u_G - 5.78\eta_{\max}S^2 - 5.6\eta_{\max}Sq_G - 7.64\eta_{\max}Su_G + 5.09\eta_{\max}q_G^2 + 3.35\eta_{\max}q_Gu_G + 7.19\eta_{\max}u_G^2 +$$
$$+7.22S^3 - 1.48S^2q_G + 13.25S^2u_G - 50.9Sq_G^2 - 20.74Sq_Gu_G + 0.49Su_G^2 - 24.57q_G^3 + 11.39q_G^2u_G + 0.47q_Gu_G^2 - 3.7u_G^3. \quad (5)$$

Як видно з табл. 2, за допомогою (5) ми можемо забезпечити точність обчислення введеного показника функціональної стійкості (1) із середньою точністю, рівною 3 %. Цю точність можна покращити до 2 % (див. табл. 2), використавши вже п'ятивимірну поліноміальну регресійну модель четвертого степеня, тобто поліном $\tilde{P}_4 = \tilde{P}_4(\eta_{\min}, \eta_{\max}, S, q_G, u_G)$. Однак, згідно з проведеними розрахунками, кількість необхідних обчислень зросте приблизно в 2.796 раз, що може бути недоречним з урахуванням можливої необхідності попереднього обчислення параметрів $\eta_{\min}$, $\eta_{\max}$, S, q_G та u_G. Як видно з рис. 2, з тієї самої причини недоречним може бути використання поліноміальної регресії ще вищих степенів. З іншого боку, якщо розглядати п'ятивимірну лінійну, квадратичну та кубічну регресію, то кубічна забезпечуватиме найкращу точність (див. табл. 2). Відповідно, якщо оцінювати показник функціональної стійкості $\overline{R}$ за допомогою три- або п'ятивимірної поліноміальної регресії, то найоптимальнішим (згідно з табл. 1, 2 та рис. 1, 2) варіантом є використання п'ятивимірної кубічної регресійної моделі, тобто полінома (5).

ISSN 2788-6603

Дискусія і висновки

Сформульовано новий структурний показник функціональної стійкості $\bar{R}$. Цей показник є числовою величиною, значення якої коливається в проміжку від 0 до 1. Характеристикою функціональної стійкості розглядуваної розподіленої інформаційної системи буде близькість цієї величини до одиниці, тобто чим вищим є значення $\bar{R}$, тим більш функціонально стійкою можна вважати розглядувану систему.

Вказаний показник функціональної стійкості обчислюється досить складно. Особливо це відчутно у випадку розподілених систем із великою кількістю машин. Тому в роботі зразу ж досліджено метод його оцінювання за класичними методами наближення, а саме за допомогою багатовимірної поліноміальної регресії.

В межах дослідження можливостей застосування багатовимірних поліноміальних регресійних моделей для оцінювання введеного показника функціональної стійкості (1) розглянуто тривимірну та п'ятивимірну поліноміальну регресію P_v та $\tilde{P}_v$, відповідно. Установлено, що збільшення степеня поліноміальної регресії може збільшити точність оцінювання показника (1) в обох випадках, однак, у випадку п'ятивимірної регресії поведінка середньої абсолютної та середньої відносної похибок більш прогнозована, а їхні значення менші. Це, у свою чергу, може забезпечити необхідність меншої кількості обчислень. Тому основну увагу в подальшому дослідженні сконцентровано саме на застосуванні п'ятивимірної поліноміальної регресії в контексті оцінювання показника функціональної стійкості (1). Враховуючи те, як змінюється середня відносна та середня абсолютна похибки для поліномів $\tilde{P}_v$ залежно від степеня v (див. рис. 1) разом із кількістю необхідних для їх застосування обчислень (див. рис. 2), установлено, що в досліджуваному випадку найдоцільнішим є застосування п'ятивимірної кубічної регресії, тобто полінома (5). Цей поліном у середньому може забезпечити точність обчислення введеного в роботі показника функціональної стійкості у 3 %.

У такий спосіб в роботі наведено новий структурний показник функціональної стійкості розподілених інформаційних систем і вказано можливий метод його оцінювання, який вимагатиме значно менше обчислень.

Внесок авторів: Олег Барабаш – огляд літератури, збір даних, формулювання основної наукової ідеї; Андрій Макарчук – розроблення методів і методології дослідження, опис результатів і написання висновків.

Список використаних джерел

Abdullayev, F. G., Ozkartepe, P., Savchuk, V. V., & Shidlich, A. L. (2019). Exact constants in direct and inverse approximation theorems for functions of several variables in the spaces Sp. *Filomat,* 33(5), 1471–1484. https://doi.org/10.2298/FIL1905471A

Барабаш, О. В. (2004). *Побудова функціонально стійких розподілених інформаційних систем.* Національна академія оборони України.

Барабаш, О. В., & Кравченко, Ю. В. (2002). Функціональна стійкість – властивість складних технічних систем. *Збірник наукових праць Національної академії оборони України,* 40, 225–229. Національна академія оборони України.

Барабаш, О. В., Обідін, Д. М., Саланда, І. П., & Макарчук, А. В. (2024). Порівняльний аналіз наближення ймовірнісного показника функціональної стійкості за допомогою поліномів Бернштейна та нейронних сіток прямого розповсюдження. *Зв'язок*, 3, 32–37. https://doi.org/10.31891/2219-9365-2024-77-6

Antonevych, M., Didyk, A., & Snytyuk, V. (2019). Optimization of functions of two variables by deformed stars method. In O. Yudin (Ed.). *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)* (pp. 475–480). Taras Shevchenko national university of Kyiv. https://doi.org/10.1109/ATIT49449.2019.9030475

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). The MIT Press.

Kravchenko, Y., Leschenko, O., & Mykus, S. (2016). Functional stability of information and telecommunication systems. *East European Scientific Journal,* 2(6), 47–52. https://doi.org/10.2298/FIL1905471A

Kovalchuk, D., Kravchenko, Y., Starkova, O., Herasymenko, K., Tarasenko, N., & Riabtsev, V. (2020). Development of recommendations for the implementation of virtualization concepts in modern networks. In D. Ageyev (Ed.). *2020 IEEE International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)* (pp. 797–802). National Aerospace university "KhAI". https://doi.org/10.1109/PICST51311.2020. 9467918

Makarchuk, A., Kal'chuk, I., Kharkevych, Y., & Kharkevych, G. (2022). Application of trigonometric interpolation polynomials to signal processing. In O. Yudin (Ed.). *2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT)* (pp. 156–159). Taras Shevchenko national university of Kyiv. https://doi.org/10.1109/ATIT58178.2022.10024182

Mashkov, O. A., Murasov, R. K., Kravchenko, Y. V., Dakhno, N. B., Leschenko, O. A., & Trush, O. V. (2021). Optimal forecast algorithm based on compatible linear filtration and extrapolation. *Mathematical Modeling and Computing,* 8(2), 157–167. https://doi.org/10.23939/mmc2021.02.157

Миронюк, М. Ю., Майстров, О. О., Мусієнко, А. П., & Макарчук, А. В. (2024). Аналіз побудови інтелектуальної інформаційної системи на основі поняття функціональної стійкості. *Зв'язок,* 1, 3–8. https://doi.org/10.31673/2412-9070.2024.010308

Shidlich, A. L. (2013). *Approximations of certain classes of functions of several variables by greedy approximants in the integral metrics.* Institute of Mathematics of NAS of Ukraine.

Snytyuk, V., Antonevich, M., & Didyk, A. (2020). Optimization of fire monitoring system using the deformed stars method. *Information Systems and Technologies Security*, 1(2), 60–66. https://doi.org/10.17721/ISTS.2020.1.60-66

References

Abdullayev, F. G., Ozkartepe, P., Savchuk, V. V., & Shidlich, A. L. (2019). Exact constants in direct and inverse approximation theorems for functions of several variables in the spaces Sp. Filomat, 33(5), 1471–1484. https://doi.org/10.2298/FIL1905471A

Antonevych, M., Didyk, A., & Snytyuk, V. (2019). Optimization of functions of two variables by deformed stars method. In O. Yudin (Ed.). *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)* (pp. 475–480). Taras Shevchenko national university of Kyiv https://doi.org/10.1109/ATIT49449.2019.9030475

Barabash, O. V. (2004). Construction of functionally stable distributed information systems. National Academy of Defense of Ukraine [in Ukrainian].

Barabash, O. V., & Kravchenko, Y. V. (2002). Functional stability – a property of complex technical systems. Collection of Scientific Papers of the National Academy of Defense of Ukraine, 40, 225–229 [in Ukrainian].

Barabash, O. V., Obidin, D. M., Salanda, I. P., & Makarchuk, A. V. (2024). Comparative analysis of the approximation of a probabilistic functional stability indicator using Bernstein polynomials and feedforward neural networks. *Sviaz*, 3, 32–37 [in Ukrainian]. https://doi.org/10.31891/2219-9365-2024-77-6

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). The MIT Press.

Kravchenko, Y., Leschenko, O., & Mykus, S. (2016). Functional stability of information and telecommunication systems. *East European Scientific Journal,* 2(6), 47–52. https://doi.org/10.2298/FIL1905471A

Kovalchuk, D., Kravchenko, Y., Starkova, O., Herasymenko, K., Tarasenko, N., & Riabtsev, V. (2020). Development of recommendations for the implementation of virtualization concepts in modern networks. In D. Ageyev (Ed.). *2020 IEEE International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)* (pp. 797–802). National Aerospace university "KhAI. https://doi.org/10.1109/PICST51311.2020. 9467918

Makarchuk, A., Kal'chuk, I., Kharkevych, Y., & Kharkevych, G. (2022). Application of trigonometric interpolation polynomials to signal processing. In O. Yudin (Ed.). *2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT)* (pp. 156–159). Taras Shevchenko National University of Kyiv. https://doi.org/10.1109/ATIT58178.2022.10024182

Mashkov, O. A., Murasov, R. K., Kravchenko, Y. V., Dakhno, N. B., Leschenko, O. A., & Trush, O. V. (2021). Optimal forecast algorithm based on compatible linear filtration and extrapolation. *Mathematical Modeling and Computing,* 8(2), 157–167. https://doi.org/10.23939/mmc2021.02.157

Миронюк, М. Ю., Майстров, О. О., Мусієнко, А. П., & Макарчук, А. В. (2024). Аналіз побудови інтелектуальної інформаційної системи на основі поняття функціональної стійкості. *Зв'язок,* 1, 3–8. https://doi.org/10.31673/2412-9070.2024.010308

Shidlich, A. L. (2013). *Approximations of certain classes of functions of several variables by greedy approximants in the integral metrics.* Institute of Mathematics of NAS of Ukraine.

Snytyuk, V., Antonevich, M., & Didyk, A. (2020). Optimization of fire monitoring system using the deformed stars method. *Information Systems and Technologies Security*, 1(2), 60–66. https://doi.org/10.17721/ISTS.2020.1.60-66

Отримано редакцією журналу / Received: 19.09.24
Прорецензовано / Revised: 22.09.24
Схвалено до друку / Accepted: 04.11.24

Oleg BARABASH, DSc (Engin.), Prof.
ORCID ID: 0000-0003-1715-0761
e-mail: bar64@ukr.net
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

Andrii MAKARCHUK, PhD Student
ORCID ID: 0000-0002-6422-7488
e-mail: andreymakarh2@gmail.com
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

DEVELOPMENT OF A NEW INDICATOR OF FUNCTIONAL RELIABILITY AND ITS EVALUATION USING MULTIVARIABLE POLYNOMIAL REGRESSION

Background. ***Functional The functional stability of distributed systems is becoming increasingly significant with the advancement of information technologies. Consequently, the formalization of this concept has gained relevance. Mathematical formalization of functional stability in the form of its indicators and criteria has been underway for several decades. Functional stability indicators play a key role in this process, and many such indicators have already been formulated. However, a major drawback of most of these indicators is that they are not only computationally complex but also dependent on numerous other parameters, or they fail to comprehensively describe the functional stability of the distributed system in question. In this paper, a new functional stability indicator is introduced that avoids the second of these drawbacks. The first drawback is addressed through the use of an estimation method based on multivariate polynomial regression.***

Methods. ***The study utilized methods of computer modeling and approximation techniques.***

Results. ***A modification of an existing indicator, known in the literature as the probability of reliability, was chosen as the method for developing the new functional stability indicator. By making certain assumptions and applying transformations, a measure was obtained that possesses certain desirable properties, namely: this measure lies strictly within the interval from zero to one, and the larger it is, the more functionally stable the distributed system under consideration can be deemed. However, the resulting functional stability indicator requires extensive calculations, prompting an attempt to estimate this indicator using approximation methods. This study explored the potential of applying multivariate polynomial regression. According to computer modeling, to achieve an average accuracy of two percent, it is sufficient to use a five-dimensional polynomial regression of the fourth degree. Increasing the degree of the five-dimensional regression model beyond this does not result in significant error reduction.***

Conclusions. ***The functional stability indicator introduced in this study provides a convenient means for investigating the functional stability of distributed systems. However, it demands a significant amount of computation. For this reason, a method for estimating the introduced functional stability indicator has been presented, which allows for relatively accurate computation of this indicator.***

Keywords: ***functional stability, indicator, approximation, optimization, regression, multivariable functions.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

УДК 004.41
DOI: https://doi.org/10.17721/AIT.2024.1.07

Олексій БИЧКОВ, д-р техн. наук, проф.
ORCID ID: 0000-0002-9378-9535
e-mail: oleksiibychkov@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

Микола МОРОЗ, асп.
ORCID ID: 0000-0001-6953-683X
e-mail: mukolamoroz@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

"MANAGER-DISPATCHER": ПАТЕРН ЗАБЕЗПЕЧЕННЯ АДАПТИВНОЇ ПОВЕДІНКИ ПРОГРАМНИХ СИСТЕМ НА ОСНОВІ ПОДІЙ

Вступ. ***Адаптивна поведінка сучасних програмних систем стає ключовим чинником їх успішного функціонування в умовах зовнішніх і внутрішніх дестабілізуючих факторів. Програми, які працюють із критичними даними або виконують важливі операції, повинні забезпечувати безперервність роботи, незважаючи на збої, атаки чи помилки. Для досягнення цієї мети пропонують різні підходи. Одним із них є застосування патернів проєктування, які дозволяють забезпечити надійність та адаптивність системи. У цій статті представлено патерн "Manager-Dispatcher", який поєднує властивості патернів "Publish-Subscribe" та "Strategy" для забезпечення адаптивної поведінки програмних систем за рахунок оброблення подій.***

Методи. ***Для розроблення патерну "Manager-Dispatcher" використано методи модульного проєктування та динамічного оброблення подій. Патерн передбачає автоматичний вибір стратегії функціонування модуля на основі подій, що відбуваються у системі. Проведено теоретичний аналіз існуючих підходів до адаптації поведінки системи і на основі цього розроблено новий патерн, який дозволяє динамічно змінювати стратегії роботи модулів у відповідь на змінні умови середовища, які визначаються подіями у системі. Розглянуто кілька гіпотетичних сценаріїв застосування для ілюстрації роботи патерну. Розроблено й описано приклад програмної системи із застосуванням патерну.***

Результати. ***Розроблений патерн "Manager-Dispatcher" дозволяє програмним модулям автоматично адаптувати стратегії функціонування на основі подій у системі. Основні переваги патерну включають модульність, розширюваність та адаптивність поведінки. Патерн може бути корисним у вбудованих системах, системах реального часу й інтерактивних інтерфейсах, де важливо забезпечити швидку та гнучку реакцію на події.***

Висновки. ***Патерн "Manager-Dispatcher" пропонує перспективний підхід до проєктування адаптивних програмних систем, орієнтованих на події. Завдяки можливості динамічної зміни стратегій функціонування, патерн забезпечує високий рівень гнучкості в умовах динамічних змін. У подальших дослідженнях планується вдосконалення патерну та розроблення інструментів для його полегшеного впровадження і тестування. Запропонований підхід сприяє побудові модульних та адаптивних систем, здатних забезпечувати стабільне функціонування навіть у складних умовах.***

Ключові слова: ***патерни проєктування, адаптивна поведінка, оброблення подій, автономні системи.***

Вступ

Актуальність дослідження обумовлена зростанням складності сучасних програмних систем і вимогою до них працювати в умовах постійно змінного середовища. Програмні системи все частіше стикаються з вимогами забезпечення гнучкості й адаптивності для підтримки безперебійної роботи під впливом внутрішніх і зовнішніх подій. У контексті швидкого розвитку технологій і збільшення обсягів оброблюваної інформації, адаптивні системи дозволяють підвищити продуктивність і знизити ризики збоїв. Тому створення патернів, які надають системам можливість адаптувати поведінку в реальному часі, є важливим напрямом досліджень, що сприяє розвитку масштабованих рішень для сучасного програмного забезпечення.

Мета цього дослідження – розробити патерн проєктування, який забезпечує адаптивну поведінку програмних систем через динамічну зміну стратегій функціонування модулів у відповідь на події та зміни умов середовища.

Завданнями цього дослідження є: провести аналіз існуючих підходів і патернів для забезпечення адаптивної поведінки програмних систем, продумати функціональність, реалізувати необхідну функціональність у новому патерні проєктування, виконати аналіз отриманих теоретичних результатів щодо розробленого патерну.

Ідея патернів проєктування отримала популярність завдяки роботі "Банди чотирьох" (Gamma et al., 1995). У книзі "Design Patterns: Elements of Reusable Object-Oriented Software" автори виклали основи використання патернів для створення повторно використовуваного об'єктно-орієнтованого програмного забезпечення.

У роботі (Bruns, & Dunkel, 2013) автори пропонують каталог патернів, які забезпечують розроблення архітектур на основі подій. Вони вводять різні патерни для структурування і створення комплексних систем оброблення подій.

У роботі (Mannava, & Ramesh, 2012) запропоновано загальний патерн для адаптивної реконфігурації автономних обчислювальних систем. Цей патерн спрямовано на автоматизацію самокерування систем, що дозволяє їм адаптуватися до змін у середовищі без залучення адміністраторів. Патерн базується на потоках вхідних даних, які аналізуються і за наявності тригерів патерн на базі правил приймає рішення про реконфігурацію. Подібні ідеї авторів також присутні у роботі (Mannava, & Ramesh, 2011), де вони описують патерн, який дозволяє автономно компонувати й адаптувати вебсервіси на основі контексту. Цей патерн є розширенням патернів Case-based Reasoning, Strategy, Observer і спрямований на забезпечення адаптивності сервісів до змінних умов.

У роботах (Ramirez, 2008) та (Ramirez, & Cheng, 2010) автори аналізували понад тридцять досліджень і проєктів, виявляючи патерни, що сприяють розробленню адаптивних систем. У результаті аналізу узагальнено й описано каталог патернів для розроблення динамічних адаптивних систем; автори також запропонували класифікацію для розроблених патернів згідно з адаптивним функціями: патерни моніторингу, патерни прийняття рішення та патерни реконфігурації.

В цій роботі ми представляємо патерн адаптивного оброблення подій "Manager-Dispatcher", який поєднує властивості патернів "Publish-Subscribe" та "Strategy" для забезпечення адаптивної поведінки системи. На відміну від традиційних підходів, наш патерн дозволяє автоматично змінювати стратегії функціонування програмного модуля залежно від

поточного контексту та стану системи. Порівняно з рішеннями у джерелах (Mannava, & Ramesh, 2011; Mannava, & Ramesh, 2012; Ramirez, & Cheng, 2010), які розглядають реконфігурацію системи як необхідну умову забезпечення адаптивності системи, ми пропонуємо змінювати стратегію функціонування роботи програмного модуля. Також патерн забезпечує подальший розвиток підходів проєктування патернів, описаний у зазначених джерелах.

Методи

Оброблення подій є фундаментальною частиною багатьох програмних систем, від вбудованих систем контролю до комплексних користувацьких інтерфейсів. Історично, патерн "Observer" (Gamma et al., 1995) надав базовий механізм для взаємодії видавців і підписників на події, дозволяючи об'єктам реагувати на зміни в інших об'єктах без прямої залежності між ними, але натомість кожному підписнику на подію варто знати про видавця подій і самостійно підписуватись на події. Гнучкішим підходом до керування подіями став патерн "Publish-Subscribe" або його менш відома версія "Event Channel" (Buschmann, 1996, pp. 339–343). Цей патерн дозволяє видавцям генерувати події, не знаючи, хто є підписниками цих подій. Підписники, у свою чергу, можуть отримувати нотифікації про ті події, які їх цікавлять, без необхідності знати про джерело подій. Саме тому "Publish-Subscribe" забезпечує високий рівень децентралізації та гнучкості.

Завдяки подіям і реакціям на них програмні модулі можуть формувати неявні слабко зв'язані ієрархічні структури залежності, тобто один модуль залежатиме від подій, що виникають в іншому модулі. Таким способом формуватиметься певна ієрархічна модульна архітектура, в якій модулі не знатимуть про модулі, які залежні від них, й орієнтуватимуться лише на події, що виникають. Зв'язки модулів на основі подій визначатимуться динамічно, хоча можуть бути запропоновані статичні підходи для формування архітектури залежності.

Під програмними модулями ми розуміємо окремі частини програмної системи, які виконують певну функцію системи, причому за рахунок підписок на події ці модулі можуть змінювати алгоритми виконання завдань і забезпечення функцій, що ми називаємо загальним терміном "зміна стратегії". Патерн "Strategy" (Gamma et al., 1995) пропонує механізм вибору поведінки об'єкта під час виконання програми, надаючи можливість змінювати алгоритми оброблення без зміни контексту їх виконання. Зміна стратегії програмного модуля може полягати у запуску процесу відновлення, зміні алгоритму роботи для збереження виконання функції програмного модуля, переході на резервні ресурси, адаптації до нових умов середовища, зміні пріоритетів оброблення задач тощо.

Обробник події не обов'язково має змінювати стратегію. Для модуля може існувати окрема множина подій, які він не здатен самостійно опрацювати і запустити відповідні дії, серед яких можуть бути усунення причин, протидії, корекції стану. Саме тому, якщо модуль зазнає невідомої або критичної помилки, що перешкоджає його подальшому функціонуванню, він також може сповістити залежні модулі про те, що цей програмний модуль є несправним, а залежні модулі мають відповідним способом змінити свої стратегії. Таким чином програмний модуль передає відповідальність за опрацювання події на ієрархічно вищі модулі.

Саме тому, використовуючи запропонований патерн, програмний модуль може бути одночасно і видавцем, і підписником. Це означає, що він має можливість як реагувати на отримані події, так і генерувати нові події у відповідь на оброблення поточних подій. Модуль може генерувати більш високорівневу помилку та повідомити про неї своїх підписників, покладаючи відповідальність за її опрацювання на них. Цей підхід нагадує концепцію каскадного оброблення подій, де оброблення однієї події може призводити до генерації інших подій, які у свою чергу обробляються іншими модулями. Це забезпечує багаторівневе й ієрархічне оброблення помилок і подій, що підвищує надійність і гнучкість системи.

Опис патерну. "Manager-Dispatcher" – це поведінковий патерн проєктування, який дозволяє компонентам системи ефективно реагувати на різноманітні події, адаптуючи свою поведінку на основі поточного контексту. В основі патерну лежить ідея про те, що обробник подій повинен мати можливість адаптувати поведінку компонента у разі виникнення певних подій, вибираючи найбільш підходящу стратегію функціонування компонента з набору доступних стратегій. Такий підхід дозволяє системі залишатися гнучкою і стійкою у складних умовах. Також цей підхід змушує розробника визначати, на які події варто реагувати компонентові, а також проєктувати відповідні стратегії поведінки.

Назва патерну відображає функціональну роль двох фахів – "менеджера", який ухвалює рішення, і "диспетчера", який виконує координацію. Обираючи цю назву, ми вкладали такий сенс: менеджер керує і робить вибір (в нашому випадку – керує / приймає події і вибирає стратегії для компонента), а диспетчер виконує конкретні дії та розподіл роботи / обов'язків (в нашому випадку – розподіл подій між компонентами та виконання стратегій).

Діаграму класів патерну "Manager-Dispatcher" наведено на рис. 1. Патерн складається з таких основних елементів:

- подія (Event): об'єкт, що представляє зміну стану, зовнішні зміни тощо; кожна подія має заздалегідь визначений тип (EventType), а також містить корисну інформацію, яка може бути необхідна компоненту-підписнику (час, видавець, причина, опис тощо);
- видавець (AnEventProducer): компонент, який генерує події; видавець не залежить і не володіє інформацією про підписників;
- підписник (Subscriber): компонент, який реагує на певні типи подій; він динамічно підписується на ті типи подій, які його цікавлять, і має змогу вибирати відповідну стратегію функціонування як реакцію на подію;
- менеджер подій (EventManager): компонент, який відповідає за координацію між видавцями і підписниками, забезпечуючи доставку події тим підписникам, які підписались на цей тип події;
- селектор стратегій (StrategySelector): визначає логіку вибору відповідної стратегії функціонування компонента на основі поточного контексту; контекстом може служити як інформація, що надана у події, так і стан системи;
- стратегія (Strategies): алгоритм функціонування компонента, який може бути вибраний підписником у результаті оброблення конкретної події, адаптація поведінки компонента до змін.

Взаємодія компонентів. На стадії ініціалізації системи кожен компонент-підписник має здійснити підписку на тип події, що його цікавить, через інтерфейс IEventSubscriptionService. Коли подія генерується в системі, EventManager нотифікує всіх підписників, які підписались на цей тип події. Кожен підписник у власному методі оброблення події може використовувати StrategySelector для вибору найкращої стратегії подальшого функціонування залежно від поточного

ISSN 2788-6603

контексту. Контекстом, наприклад, може бути стан системи та/або стан зовнішнього середовища. Після вибору стратегії підписник продовжує функціонувати з адаптованою поведінкою.

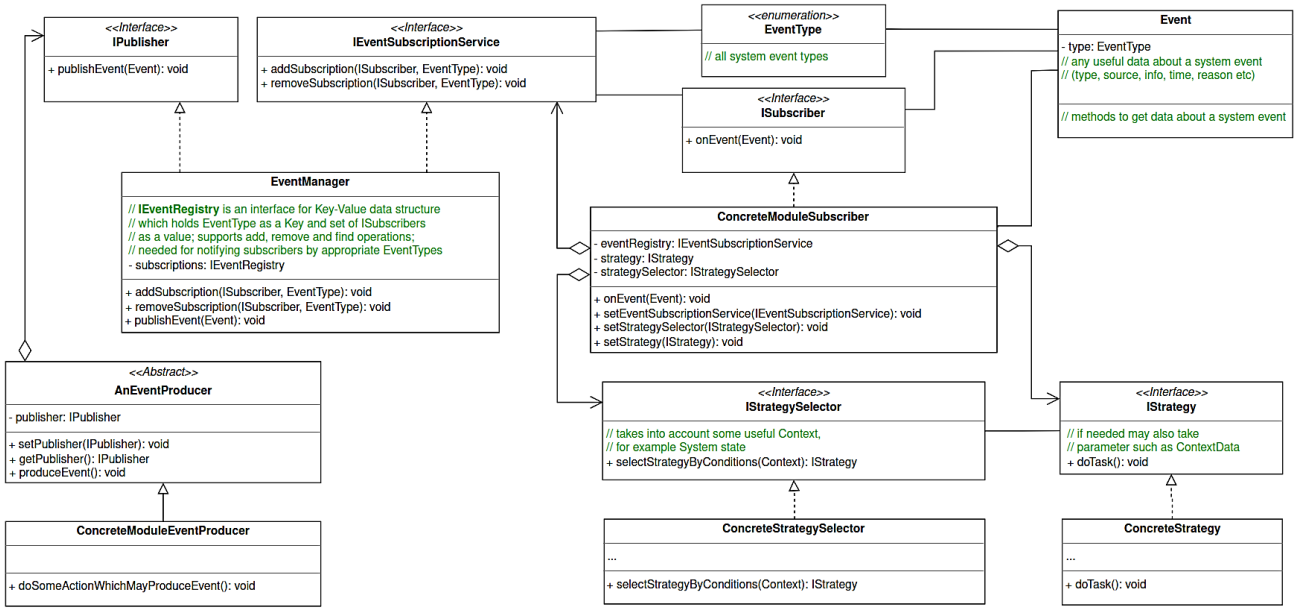


Рис. 1. UML-діаграма класів патерну "Manager-Dispatcher"

Цей патерн забезпечує високий рівень модульності та розширюваності, дозволяючи легко додавати нові стратегії, селектори стратегій, підписників, видавців і події без необхідності зміни вже існуючого коду. Він також сприяє відокремленню відповідальності – кожен компонент фокусується на своєму конкретному завданні, знижуючи складність системи та полегшуючи тестування й підтримку.

Для ілюстрації процесу оброблення подій у патерні на рис. 2 наведено діаграму послідовності. Ця діаграма демонструє взаємодію між основними компонентами патерну. Вона відображає послідовність кроків, що відбуваються від моменту генерації події до її оброблення підписником, при цьому оброблення події може полягати і у виборі найкращої стратегії подальшого функціонування, і у нотифікації підписників, що підписані на події, створювані поточним підписником.

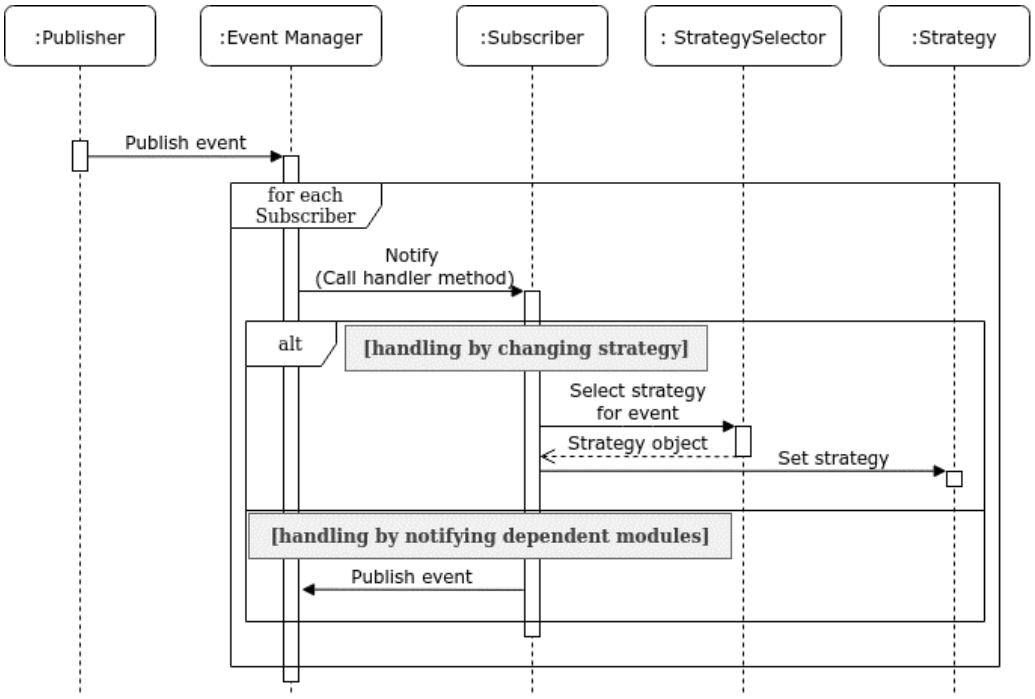


Рис. 2. UML-діаграма послідовності оброблення подій у патерні "Manager-Dispatcher"

Приклади використання. Щоб продемонструвати можливості патерну "Manager-Dispatcher", розглянемо кілька гіпотетичних сценаріїв його застосування:

1. Вбудовані системи реального часу. У системах реального часу, таких як контролери промислового обладнання, важливо швидко реагувати на зміни стану сенсорів та інших входів. Використовуючи наш патерн, система може автоматично змінювати стратегії залежно від поточного стану обладнання, забезпечуючи безперервність роботи та мінімізуючи ризик збоїв.

2. Інтерактивні інтерфейси. У інтерактивних інтерфейсах користувача, де необхідна гнучка реакція на дії користувачів, наш патерн дозволяє швидко адаптувати стратегії функціонування для забезпечення плавної та ефективної роботи системи. Наприклад, інтерфейс може автоматично змінювати спосіб відображення даних у відповідь на зміну умов навантаження або вимог користувача.

3. Системи моніторингу. У системах моніторингу, таких як системи безпеки, патерн використовують для автоматичної адаптації до нових загроз. Наприклад, у разі виявлення підозрілої активності система може змінити стратегії реагування – активувати додаткові заходи безпеки або перенаправити трафік.

Ми вважаємо, що патерн буде особливо корисним для вбудованих систем, оскільки вони зазвичай функціонують у межах однієї програми (тобто не мають ОС і різних процесів, які потрібно синхронізувати між собою), а також потребують гнучкості та необхідності реагування на події середовища.

Результати

Теоретичні результати (оцінювання патерну), що представлені у статті, реалізовано для побудови простої системи для трекінгу цілі. Приклад створено з використанням мови C++. У прикладі відбувається аналіз зображення з відеопотоку та пошук об'єкта. Критерій детектування об'єкта – це наявність червоного кольору і розмір не менше ніж 50 x 50 пікселів. Детектування об'єкта відбувається за окремим алгоритмом, а коли об'єкт буде знайдено на зображенні, то буде застосовано алгоритм трекінгу об'єкта. У прикладі патерн керує подіями в описаній програмній системі; у разі виникнення події відбувається адаптування програмної системи до події, що виникла.

Описана система має такі модулі: модуль аналізу зображення, модуль-стратегія детектування об'єкта, модуль-стратегія трекінгу об'єкта, модуль відео провайдера, модуль менеджера подій (клас EventManager), модуль нотифікацій. Модуль аналізу зображення та модуль нотифікацій є підписниками на події. В системі є два типи подій: "ціль знайдено" та "ціль втрачено". За наявності цих подій модуль аналізу зображення змінює свою стратегію функціонування, а саме: за замовчуванням модуль аналізу зображення має стратегію аналізу – детектування; якщо стратегія детектування знаходить об'єкт, то генерується відповідна подія, а модуль аналізу зображення змінює стратегію аналізу на трекінг; якщо стратегія трекінгу детектує втрату цілі, то генерується відповідна подія і модуль аналізу зображення змінює стратегію аналізу на детектування. Модуль нотифікацій підписується на обидві події і пише інформацію про отриману подію у консоль.

Наступний код демонструє роботу запропонованого патерну.

Файл *main.cpp*:

```
int main()
{
	cv::Mat image;
	cv::namedWindow("Video Player");
	app::VideoProvider video_provider{};

	core::EventManager event_manager{};
	core::AnEventProducer event_producer{&event_manager};
	app::ImageAnalysisStrategySelector strategy_selector{&event_producer};
	// модуль аналізу зображення
app::ImageAnalyzer image_analyzer{&event_manager, &strategy_selector};
	// модуль нотифікацій
	app::Notifier logger{&event_manager};

	while (true) {
	if (!video_provider.getFrame(image)) { break; }

	// зміна стратегій оброблення кадрів відбувається всередині модуля аналізу зображення
	if (!image_analyzer.processFrame(image))
	std::cout << "Image analysis module could not process frame!" << std::endl;

	cv ::imshow("Video Player", image);
	}
	return 0;
}
```

image_analyzer – це об'єкт, який відповідає за аналіз зображення, поведінку якого було описано вище. Його обробник подій та метод оброблення кадру виглядає так:

```
void ImageAnalyzer::onEvent(core::Event event)
{
    strategy_ = strategy_selector_->selectStrategyForEvent(event);
}
    bool ImageAnalyzer::processFrame(cv::Mat& image)
{
    if (!strategy_) { return false; }
    return strategy_->process(image);
}
```


ISSN 2788-6603

А метод для вибору стратегії:

```
core::IImageAnalysisStrategy* ImageAnalysisStrategySelector::selectStrategyForEvent(core::Event event)
{
    switch (event.getType()) {
    case core::EventType::OBJECT_FOUND:
    return new Tracker(event_producer_, *((cv::Rect*)event.getData()));
    case core::EventType::OBJECT_LOST:
    return new Detector(event_producer_);
    default:
    return getDefaultStrategy();
}
}
```

Клас нотифікатора є дуже простим і ось так виглядає його конструктор (де відбувається підписка на події) та обробник подій:

```
Notifier::Notifier(core::IEventSubscriptionService* event_subscription_service)
{
    event_subscription_service->addSubscription(this, core::EventType::OBJECT_FOUND);
    event_subscription_service->addSubscription(this, core::EventType::OBJECT_LOST);
}

void Notifier::onEvent(core::Event event)
{
    switch (event.getType()) {
    case core::EventType::OBJECT_FOUND:
    std::cout << "Got an event with a type: OBJECT_FOUND." << std::endl;
    break;
    case core::EventType::OBJECT_LOST:
    std::cout << "Got an event with a type: OBJECT_LOST." << std::endl;
    break;
    default:
    break;
}
}
```

Отже, з прикладу можна побачити, що модуль аналізу зображення демонструє адаптацію поведінки у відповідь на подію; цей компонент працює на основі стратегій і здатний змінювати їх за логікою окремого селектора стратегій, що є гнучким і розширюваним рішенням. Зауважимо, що модуль аналізу зображення нічого не знає про окремі стратегії. Модулі у цьому прикладі зв'язані подіями, що також сприяє гнучкості й подальшому масштабуванню.

З повною версією прикладу можна ознайомитись у роботі М. Мороза (https://github.com/mr-holodok/director-dispatcher-pattern-example).

Узагальнюючи отримані результати, архітектура патерну "Manager-Dispatcher" демонструє такі властивості:

- гнучкість реакції: система може автоматично адаптуватися до змін у середовищі, що значно підвищує її гнучкість;
- модульність, розширюваність і масштабованість патерну;
- зниження залежності між компонентами: завдяки централізації керування подіями через EventManager, зменшується зв'язність між видавцями та підписниками, що спрощує модифікацію та розширення системи;
- підвищення відмовостійкості: динамічний вибір стратегій дозволяє системі краще протистояти помилкам або збоям, адаптуючи поведінку підписників залежно від поточного стану системи та змін середовища.

Наукова цінність дослідження полягає у тому, що розроблено новий патерн проєктування, який забезпечує адаптивну поведінку програмних модулів у відповідь на події, що виникають у програмній системі. Ця функціональність може використовуватись для розв'язання таких практичних задач: автоматичного налаштування програмних компонентів під поточні умови роботи, підвищення надійності систем завдяки швидкому реагуванню на зовнішні та внутрішні зміни, а також для створення масштабованих і модульних рішень у вбудованих системах, системах реального часу й інтерактивних інтерфейсах.

Дискусія і висновки

Розроблений патерн "Manager-Dispatcher" є одним з інструментів для створення адаптивних і гнучких програмних систем, але він не є "срібною кулею", яка розв'яже всі можливі проблеми. Як і будь-який інший патерн, він має свої обмеження та вимоги, які слід враховувати за його впровадження.

- Обмеження патерну:
- складність впровадження: впровадження цього патерну може вимагати значних зусиль і часу, особливо для великих і складних систем;
- підтримка та тестування: динамічна природа патерну може ускладнити процес тестування та відлагодження системи;
- непридатність для всіх сценаріїв: патерн може бути неефективним і громіздким для систем, де зміни умов і контексту відбуваються зрідка або де переважає стабільне середовище.

Патерн "Manager-Dispatcher" закладає основу для побудови систем, орієнтованих на події, що змушує архітекторів приділяти значну увагу можливим подіям та реакціям на них. Це сприяє більш гнучкому та масштабованому підходу до проєктування, де кожен модуль може адаптуватися до змін у реальному часі.

Однак важливо пам'ятати, що цей патерн не розв'язує всіх проблем. Він вимагає ретельного планування та розуміння специфіки системи, в якій він буде використовуватися. Архітектори та розробники повинні враховувати можливі події та відповідні реакції на них, що може збільшити складність початкового етапу розроблення.

У цій статті ми представили патерн "Manager-Dispatcher", який поєднує властивості патернів "Publish-Subscribe" і "Strategy" для забезпечення адаптивності поведінки програмних систем. Наш підхід дозволяє динамічно змінювати

стратегії функціонування програмного модуля у відповідь на змінювані умови середовища, забезпечуючи гнучкість та адаптивність поведінки системи.

Ми також розглянули кілька гіпотетичних сценаріїв застосування патерну разом із прикладом використання патерну; наведені результати демонструють потенціал патерну для підвищення гнучкості різних типів систем. Застосування цього патерну сприяє побудові модульних, орієнтованих на події систем, що змушує архітекторів приділяти значну увагу можливим подіям і реакціям на них.

Подальші дослідження можуть зосередитися на покращенні патерну, а також на розробленні інструментів і методологій для полегшення його впровадження та тестування.

Патерн "Manager-Dispatcher" пропонує перспективний підхід до проєктування адаптивних систем, що можуть забезпечити гнучкість у динамічних умовах сучасного програмного забезпечення.

Внесок авторів: Олексій Бичков – концептуалізація, написання – перегляд і редагування; Микола Мороз – аналіз джерел, підготовка огляду літератури, проєктування патерну, написання – оригінальна чернетка.

Список використаних джерел

Мороз, М. (2024). Приклад використання патерну "Manager-Dispatcher". https://github.com/mr-holodok/director-dispatcher-pattern-example

Bruns, R., & Dunkel, J. (2013). Towards pattern-based architectures for event processing systems. In R. Buyya, N. Horspool, A. Wellings (Eds.), Software: *Practice and Experience*, 44(11), 1395–1416. https://doi.org/10.1002/spe.2204

Buschmann, F., Sommerlad, P., Stal, M., Rohnert, H., & Meunier, R. (1996). *Pattern-Oriented software architecture volume 1. A system of patterns.* Wiley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object oriented software.* Addison-Wesley.

Mannava, V., & Ramesh, T. (2011). A Novel Event Based Autonomic Design Pattern for Management of Webservices. In D. C. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, D. Nagamalai (Eds.) *Advances in Computing and Information Technology*. *Communications in Computer and Information Science*. 198 (pp. 142–151). Springer. https://doi.org/10.1007/978-3-642-22555-0_16

Mannava, V., & Ramesh, T. (2012). A novel adaptive re-configuration compliance design pattern for autonomic computing systems. In E. P. Sumesh (Ed.). *Procedia Engineering*, 30, 1129–1137. https://doi.org/10.1016/j.proeng.2012.01.972

Ramirez, A. J. (2008). *Design patterns for developing dynamically adaptive systems* [Master's thesis]. Michigan State University. https://doi.org/doi:10.25335/ttfn-qx40

Ramirez, A. J., & Cheng, B. H. C. (2010). Design patterns for developing dynamically adaptive systems. *In the 2010 ICSE workshop on Software Engineering for Adaptive and Self-Managing Systems* (pp. 49–58). ACM Press. https://doi.org/10.1145/1808984.1808990

References

Bruns, R., & Dunkel, J. (2013). Towards pattern-based architectures for event processing systems. In R. Buyya, N. Horspool, A. Wellings (Eds.), Software: *Practice and Experience*, 44(11), 1395–1416. https://doi.org/10.1002/spe.2204

Buschmann, F., Sommerlad, P., Stal, M., Rohnert, H., & Meunier, R. (1996). *Pattern-Oriented software architecture volume 1. A system of patterns.* Wiley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object oriented software.* Addison-Wesley.

Mannava, V., & Ramesh, T. (2011). A Novel Event Based Autonomic Design Pattern for Management of Webservices. In D. C. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, D. Nagamalai (Eds.) *Advances in Computing and Information Technology*. *Communications in Computer and Information Science*. 198 (pp. 142–151). Springer. https://doi.org/10.1007/978-3-642-22555-0_16

Mannava, V., & Ramesh, T. (2012). A novel adaptive re-configuration compliance design pattern for autonomic computing systems. In E. P. Sumesh (Ed.). *Procedia Engineering*, 30, 1129–1137. https://doi.org/10.1016/j.proeng.2012.01.972

Moroz, M. (2024). An example of using the "Manager-Dispatcher" pattern [in Ukrainian]. https://github.com/mr-holodok/director-dispatcher-pattern-example

Ramirez, A. J. (2008). *Design patterns for developing dynamically adaptive systems* [Master's thesis]. Michigan State University. https://doi.org/doi:10.25335/ttfn-qx40

Ramirez, A. J., & Cheng, B. H. C. (2010). Design patterns for developing dynamically adaptive systems. In *the 2010 ICSE workshop on Software Engineering for Adaptive and Self-Managing Systems* (pp. 49–58). ACM Press. https://doi.org/10.1145/1808984.1808990

Отримано редакцією журналу / Received: 22.09.24
Прорецензовано / Revised: 01.10.24
Схвалено до друку / Accepted: 07.11.24

Oleksii BYCHKOV, DSc (Engin.), Prof.
ORCID ID: 0000-0002-9378-9535
e-mail: oleksiibychkov@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Mykola MOROZ, PhD Student
ORCID ID: 0000-0001-6953-683X
e-mail: mukolamoroz@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

"THE MANAGER-DISPATCHER": A DESIGN PATTERN FOR ENSURING ADAPTIVE BEHAVIOR OF EVENT-DRIVEN SOFTWARE SYSTEMS

Background. ***Adaptive behavior in modern software systems is becoming a key factor in their successful operation under external and internal destabilizing influences. Programs that work with critical data or perform essential operations must ensure continuity of service, despite failures, attacks, or errors. Various approaches are proposed to achieve this goal, one of which is the application of design patterns that ensure system reliability and adaptability. This paper presents the "Manager-Dispatcher" pattern, which combines the features of the "Publish-Subscribe" and "Strategy" patterns to enable adaptive behavior in software systems through event processing.***

Methods. ***The development of the "Manager-Dispatcher" pattern was based on modular design and dynamic event processing methods. The pattern provides an automatic strategy selection for module operation based on events occurring within the system. A theoretical analysis of existing approaches to adaptive behavior in systems was conducted, leading to the creation of a new pattern that enables dynamic strategy changes in modules in response to environmental changes determined by system events. Several hypothetical application scenarios were considered to illustrate the pattern's functionality, and an example software system utilizing the pattern was developed and described.***

Results. ***The developed "Manager-Dispatcher" pattern allows software modules to automatically adapt their operational strategies based on system events. Key advantages of the pattern include modularity, extensibility, and adaptive behavior. The pattern may be particularly useful in embedded systems, real-time systems, and interactive interfaces, where fast and flexible responses to events are essential.***

Conclusions. ***The "Manager-Dispatcher" pattern offers a promising approach to the design of event-driven adaptive software systems. With its ability to dynamically change operational strategies, the pattern ensures a high degree of flexibility in dynamic environments. Future research will focus on improving the pattern and developing tools to facilitate its implementation and testing. This proposed approach supports the development of modular and adaptive systems capable of maintaining stable operation even under complex conditions.***

Keywords: ***design patterns, adaptive behavior, event processing, autonomous systems.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

УДК 004.75
DOI: https://doi.org/10.17721/AIT.2024.1.08

Олександра ДМИТРЕНКО, асп., асист.
ORCID ID: 0009-0009-4785-4226
e-mail: o_dmytrenko@lll.kpi.ua
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

Марія СКУЛИШ, д-р техн. наук, проф.
ORCID ID: 0000-0002-5141-1382
e-mail: mskulysh@gmail.com
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", Київ, Україна

ГРУПУВАННЯ МІКРОПОСЛУГ ДЛЯ ВИКОРИСТАННЯ НЕПОВНО ЗАЛУЧЕНИХ РЕСУРСІВ

В с т у п . ***Завдання оптимізації використання ресурсів хмарних систем має високий пріоритет, оскільки ця технологія стала широко розповсюдженою та легкодоступною. Поширеною практикою є створення ефективних сервісів, які легко адаптуються до різних типів продуктів, наприклад, безсерверні (serverless) технології, пропрієтарні бази даних і програмне забезпечення, оптимізоване для хмари. Тестування продуктивності хмарних застосунків стало значно простішим із появою Docker і Kubernetes, що додатково збільшило попит на хмарні ресурси. Оскільки значна кількість ресурсів витрачається на розроблення й обслуговування хмарної інфраструктури, важливо розуміти, як ці ресурси можна оптимізувати.***

М е т о д и . ***Основна ідея полягає в реорганізації хмарної інфраструктури так, щоб програмне забезпечення розподілялося по серверних групах, а не серверах, за принципом комплементарності, що означає наближення навантаження на апаратне забезпечення до повного. Для формування комплементарних груп використовують методи кластеризації (K-Means), задача про множинний рюкзак, жадібний алгоритм, сортування, бінарний пошук, перемішування та поділ на основі середнього значення та стандартного відхилення. Додатково для візуалізації пропонують алгоритми FastDTW та Z-масштабування.***

Р е з у л ь т а т и . ***Розглянуто ключові характеристики комп'ютерних систем, які можна виміряти та на які можна впливати. Ці характеристики включають пропускну здатність каналів передачі, оцінку продуктивності на основі затримки, обсяг оперативної та постійної пам'яті, обчислювальну потужність і кількість ядер. Запропоновано алгоритм для визначення комплементарних екземплярів мікросервісів, які могли б ефективно використовувати серверні ресурси. Спочатку екземпляри мікросервісів класифікують за їх місткістю ключового ресурсу.***

В и с н о в к и . ***Запропоновано алгоритм, який оптимізує використання ресурсів хмарної системи за допомогою ефективного розподілу навантаження доповнених мікросервісів між різними серверними групами. Оптимізація полягає в максимізації застосування серверних ресурсів шляхом заповнення його вільного часу іншими мікросервісами й, отже, зменшенням простою серверів. Це, своєю чергою, сприятиме покращенню загальної продуктивності хмари та зменшенню витрат на обслуговування хмарних інфраструктур. Наведений підхід застосовують як до мікросервісів, так і до монолітів, з описаними мінімальними змінами. Цей алгоритм може бути корисним для постачальників хмарних послуг та організацій, які використовують хмарні середовища для розгортання своїх застосунків.***

К л ю ч о в і с л о в а : ***групування мікросервісів, процесорна ефективність, оптимізація використання енергії, хмарні системи, мікросервісна архітектура.***

Вступ

Еволюція хмарних технологій започаткувала новий виток у розробленні програмного забезпечення на початку 2000-х рр. Справжній прорив у розробленні та тестуванні хмарних застосунків стався 2013 р. із появою Docker. Упровадження технології контейнеризації, а згодом і системи оркестрації Kubernetes, створило сприятливе середовище для розвитку мікросервісної архітектури, надавши можливість ефективно моделювати роботу та масштабування мікросервісів у хмарному середовищі.

Мікросервісна архітектура демонструє особливу ефективність у масштабних застосунках (Aspire Systems, 2023), де різні функціональні компоненти можуть мати різний ступінь взаємозалежності. Характерною особливістю користувацької поведінки є нерівномірність використання різних функцій застосунку – деякі компоненти можуть потребувати значно більше ресурсів, ніж інші. Мікросервісний підхід дозволяє здійснювати вибіркове масштабування, запускаючи додаткові екземпляри лише тих компонентів, які зазнають підвищеного навантаження. Водночас і монолітні застосунки можуть запускатись у кількох екземплярах, що сприяє перерозподілу навантаження та полегшує питання обслуговування системи. Такий підхід забезпечує економічну ефективність, оскільки точкове масштабування окремих компонентів потребує менше ресурсів порівняно з розгортанням додаткових копій усього застосунку у використанні балансування навантаження (Sharma, 2023).

Для забезпечення надійності хмарної інфраструктури, оновлення серверного обладнання зазвичай планують відповідно до закінчення гарантійного терміну. Це створює передумови для максимально ефективного використання наявних обчислювальних ресурсів протягом усього життєвого циклу обладнання, адже інакше воно не окупиться.

У цій статті запропоновано алгоритм, спрямований на оптимізацію розподілу навантаження між мікросервісами й ефективне використання серверних і мережних ресурсів хмарної інфраструктури.

Типовим для серверних застосунків є нерівномірне навантаження, що залежить від специфіки застосунку та часових патернів використання. Передбачуваність цих патернів створює можливість для оптимізації – формування серверних груп або кластерів, де комбінація різних мікросервісів забезпечує близьке до оптимального сумарне навантаження. Як результат – підвищення енергетичної та обчислювальної ефективності хмарних систем впровадженням алгоритму пошуку комплементарних навантажень для роботи на серверних групах.

Постановка задачі. Розробити алгоритм для завантаження максимальної кількості серверних ресурсів хостингової системи (хмарного провайдера) через оптимізацію кількості виконуваних задач шляхом пошуку доповнень до мікросервісів і монолітів (як підмножини мікросервісів), що необхідно розмістити на серверах.

Мета дослідження. Продемонструвати можливість повного ефективного завантаження серверів, що сприятиме зменшенню їхньої кількості. Спрямувати підхід на досягнення максимальної та збалансованої завантаженості всіх компонентів системи.

Методи

В цьому розділі представлено алгоритм для пошуку комплементарних або доповнювальних навантажень, щоб кілька елементів програмного забезпечення (ПЗ) могли сформувати повне навантаження на серверну групу, таким способом використовуючи максимально повно її ресурс. На створення алгоритму надихнула нечітка теорія ґраток, зокрема поняття доповняльних ґраток (Dmytrenko, & Skulysh, 2024a) та лінійне програмування. Елементи ПЗ, мікросервіси чи моноліти, для спрощення будуть надалі називатись мікросервісами, адже і до монолітів можна ставитись як до мікросервісів за використання скейлінгу, що охоплює запуск додаткових екземплярів з балансуванням навантаження між ними. У випадках, коли поведінка мікросервісів і монолітів відрізнятиметься, слідуватиме додаткове пояснення.

Алгоритм наведено по пунктах, у деяких пунктах є посилання на інші, тому номери допоможуть зорієнтуватись.

1. **Аналіз вхідних мікросервісів.** Вхідні дані: статистика стосовно роботи кожного мікросервісу або моноліта, який сприйматиметься як мікросервіс. Статистика містить такі дані:
   - CPU (очікуване завантаження в інтервал часу, кількість ядер та їхня частота);
   - RAM (очікуване завантаження та загальна необхідна кількість);
   - канальний ресурс – channel (інтервальний очікуваний об'єм трафіка і загальна пропускна здатність каналу);
   - базові характеристики техніки, на якій виконувались випробування, такі як: кількість ядер, вимоги до жорсткого диска;
   - прапорець можливості поділу чи реплікації вхідного елемента ПЗ canBeChunked;
   - географічна зона для роботи мікросервісу. Вона може задаватись на етапі вхідних даних замовником, або ж вираховуватись у процесі роботи ПЗ на клауд-системі й оновлюватись на наступних ітераціях алгоритму.

2. **Нормалізація вхідних даних.** Перед початком роботи з ПЗ, яке прийшло з іншої системи, варто перевести метрики з іншого середовища на підходящі серверні потужності клауд-провайдера.

3. **Поділ великозатратних елементів ПЗ.** Поділ повного необхідного ресурсу на менші можливий для мікропроцесорів, а також переважно можливий і для монолітів. Для цього в початковій умові може задаватись параметр "canChunk", тобто можливість поділу. На практиці "поділом" називають scaling – масштабування, тобто створення додаткових екземплярів мікросервісів за надмірного навантаження або ж дублювання монолітів. Для розв'язання задачі ефективного використання ресурсів, зручніше оперувати поняттям повного та часткового навантаження, тому замість "розширення" використовують слово "поділ".

Поділ здійснюють на основі найбільшого, тобто ключового показника (CPU, RAM чи канальний ресурс). Жоден із показників не має перевищувати дозволеної норми. Після поділу ключовим показником стає новий найбільший показник. Навантаження мікросервісу в час його найбільшого піка має бути меншим за граничне значення

$$P_{t_{\max}} \le P_{boundary} - P_{boundary} \times \alpha_{boundary},$$

де $\alpha_{boundary}$ – прийнятний коефіцієнт недоповненості навантаження, яке може бути рівним $D_{c_{\max}}$ (див. пункт 9), тобто нехай понад 70–80 %, то до нього буде важко знайти доповнювальний. Такий мікросервіс потрібно теж помітити як габаритний.

4. **Систематизація всіх мікросерверів, що потребують серверного розміщення.** Поділ може відбуватись на основі активності роботи в межах часових проміжків. Створення додаткового екземпляра відбуватиметься тільки на певний період часу, коли буде перевищено максимальне допустиме значення за якоюсь характеристикою з пункту 1:

$$P_{t_{\max}} + \alpha_{reserve} \cdot P_{t_{\max}} \ge P_{boundary}.$$

Поділ відбуватиметься на стільки частин m, скільки повних максимальних завантажень з урахуванням безпечного резерву $\alpha_{reserve}$ може вміститись у заданий елемент, плюс те навантаження, яке не буде повним, але залишковим від ділення, округлений до більшого:

$$m = \left\lceil \frac{P_{\max} + m \cdot \alpha_{reserve}}{P_{boundary}} \right\rceil.$$

У випадках, коли той самий мікросервіс необхідно запустити на кількох географічних зонах, він не буде сприйматись як той самий мікросервіс із погляду алгоритму. Якщо однаковий мікросервіс деплоїться (інсталюється) на кілька географічних зон, щоб швидше забезпечувати відповідь клієнтам, розташованим у цих зонах, попередньо необхідно зробити заміри використання потужностей, що будуть відповідати цим зонам. Такі заміри можуть бути наведені у вхідних даних або обчислені на наступних ітераціях клауд-провайдером.

На виході: поділені мікросервіси з новими ідентифікаторами, розподілено за географічними зонами. Кожна географічна зона містить такий список груп:

1) список мікросервісів вільного розподілу, тобто мікросервіси, які можна комбінувати з будь-якими іншими;

2) список списків мікросервісних груп, які мають розміщуватись разом на підставі використання спільного ресурсу або питань безпеки;

3) список мікросервісів, які вимагають окремого серверного ресурсу, адже їхні потреби завеликі, навантаження непередбачуване, або через питання безпеки.

5. **Нормалізація шаблона роботи мікросервісу.** Нормалізація виконується через Z-Score алгоритм (Ozdemir, & Susarla, 2018) з метою пошуку класів еквівалентності. Ідея полягає в записі статистики використання мікросервісом кожного ресурсу у проміжку, який приблизно рівний [–2; 2]. Середні значення показників розташуються близько нуля, менші за середні – у від'ємній площині, а більші за середні – у додатній. Якщо показник дуже відрізняється від інших, може бути сенс поділити цей мікросервіс на 2 чи більше, щоб створювати виділений екземпляр на екстремальний час

ISSN 2788-6603

для оброблення підвищеної кількості запитів. В разі проблем із підбору доповнювального мікросервісу, це буде зроблено на пізньому етапі алгоритму (Dmytrenko, Skulysh, Globa, 2024).

6. **Візуалізація шаблонів роботи.** Як опційний крок можна розглядати візуалізацію результату порівняння схожості зразків роботи мікросервісів, прораховану за допомогою FastDTW (Salvador, & Chan, 2004) алгоритму, який накладає часові ряди в оптимальному місці (Dmytrenko, Skulysh, Globa, 2024).

7. **Пошук подібних шаблонів роботи за ключовим критерієм.** Наступним кроком є пошук схожих шаблонів роботи ПЗ. Основною задачею пошуку схожості шаблонів є виділення класів, подібних за стилем роботи мікросервісів, користуючись ключовим (найвитратнішим) критерієм. Переважно це CPU, а канальний ресурс і RAM перевіряють додатково як ті, що не накладатимуться. Якщо ж для частини мікросервісів ключовий критерій інший, то їх варто обробляти окремо за тим самим принципом.

Для пошуку схожості шаблонів використовують алгоритм кластеризації KMeans. За цим алгоритмом формують центроїди – точки, які представляють центри груп мікросервісів схожої поведінки. Оскільки статистика використання мікросервісу включає значення за певною характеристикою за цілий день, то і точка-центроїд теж являє собою часовий ряд значень обраної характеристики.

8. **Двовимірна візуалізація кластерів часових рядів.** З метою візуалізації можна представити часовий ряд як двовимірну точку користуючись методом головного компонента – Principal component analysis (PCA). Таке представлення дасть можливість побачити групи дочок та їхні центроїди та зробити висновки про якість зон скупченості. На рис. 1 ліворуч за допомогою PCA зображено поділ мікросервісів, зазначених у пункті 11, на кластери, де центроїд позначено хрестиком, а мікросервіси – точками. Праворуч показано ті самі часові ряди у звичайному вигляді. Пунктирною лінією позначені центроїди. Детальнішу інформацію можна прочитати в роботі (Dmytrenko, Skulysh, Globa, 2024).

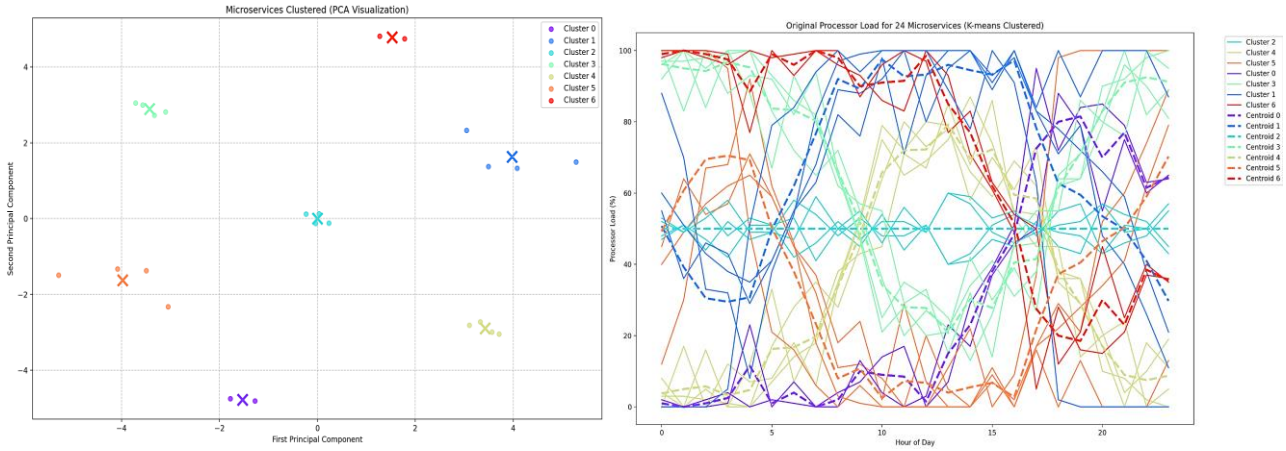


Рис. 1. Ліворуч: поділ мікросервісів на кластери методом K-Means і зображення у двовимірному вигляді за допомогою PCA методу. Праворуч: поділ тих самих кластеризованих мікросервісів методом K-Means у вигляді часових рядів

9. **Пошук доповняльних груп.** Кульмінаційним моментом алгоритму є пошук доповняльних груп мікросервісів і подальший пошук доповнювальних мікросервісів (пункт 10).

За пошуку доповнювальних груп вираховують показник розбіжності центроїдів $D_{ci,j}$ за ключовим параметром, як нормовану різницю між максимальним завантаженням релевантної серверної групи $sum_{\max_{i,j}}$ та сумою навантажень, яке зроблять обидва центроїди.

$$D_{ci,j} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{sum_{\max_{i,j}} - \left(vector_i + vector_j\right)}{sum_{\max_{i,j}}}\right).$$

Якщо результуюче значення роозміщене в межах заданої похибки $D_{ci,j} \le D_{\max_{i,j}}$, то групи вважають доповняльними. Результатом цього кроку є список проранжованих доповнень до кожного кластера.

10. **Пошук доповнювальних пар мікросервісів.** При здійсненні пошуку доповнених мікросервісів використовуються початкові метрики кожного екземпляру, а не нормовані. З двох найбільш доповнювальних груп береться по одному мікросервісу *i* та *j* з найближчою різницею в амплітуді. Щоб не перевіряти заздалегідь неробочі варіанти, можна порівняти різницю хвилі потужності, вирахувану як різниця потужностей у час максимуму $t_{\max}$ та час мінімуму $t_{\min}$, та переконатись, що вона знаходиться в певній умовою заданій межі $\Delta_{similarity}$:

$$\left(P_{i,t_{\max}} - P_{i,t_{\min}}\right) - \left(P_{j,t_{\max}} - P_{j,t_{\min}}\right) \le \Delta_{similarity}.$$

Мікросервіси накладають, як на рис. 2, та оцінюють сумарну невідповідність (Discrepancy) $D_{i,j}$ за кожним показником, тобто кількості недовикористаних ресурсів. Якщо цей показник менший за граничний, заданий у вхідній умові, за ключовим показником (напр., D_{CPU} – невідповідність, дозволена для CPU), а також нема перетинів за іншими показниками (канальний ресурс і RAM), то вибрані мікросервіси формують групу. Вони видаляються з груп пошуку доповнювальних мікросервісів, адже стають закритими для пошуку іншої пари.

Під їхню комбінацію буде виділено серверну групу, загальний ресурс якої P_{group_g} рівний їхньому максимальному ресурсу $P_{\max_g}$ з урахуванням заданої початковою умовою похибки $\alpha_{reserve}$ (alpha) – відсоток запасу ресурсу. Числове значення обчислюють як відсоток від сумарного найбільшого завантаження в одиницю часу t. У наступній формулі i та j – індекси мікропроцесорів, які входять у групу. Ця формула валідна для довільної кількості елементів у групі g.

$$P_{\max_g} = \max\left(\sum_{g \in i,j} P_g(t)\right).$$

$$P_{group_g} = P_{\max_g} + \alpha_{reserve} \cdot P_{\max_g}.$$

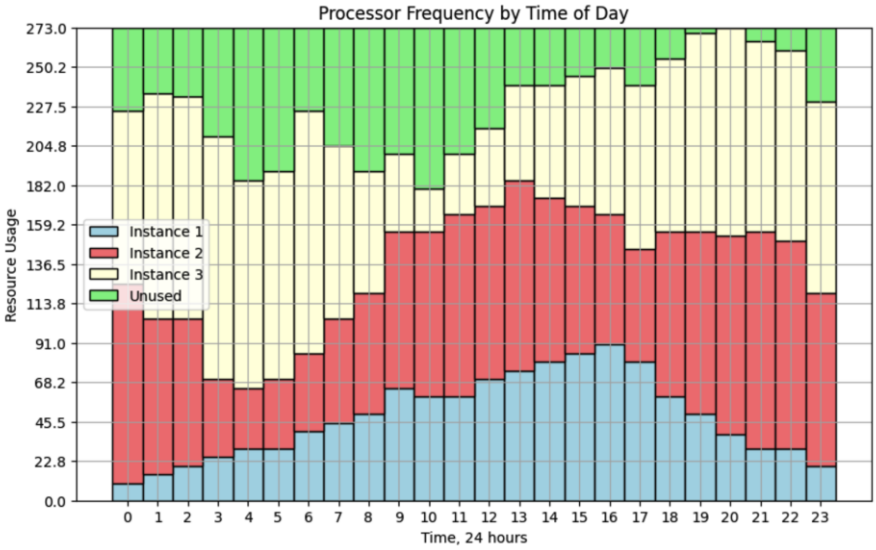


Рис. 2. Навантаження трьох мікросервісів (знизу). Порожній ресурс (зверху)

Якщо ж невідповідність $D_{i,j}$ більша за максимально дозволену $D_{g\ \max}$, обидва мікросервіси продовжують шукати собі пару, помічаючи пройдені мікросервіси як непідходящі.

Пошук пропонується здійснювати після сортування початкової групи мікросервісів, а також використовувати бінарний пошук.

Якщо в межах найкращої доповнювальної групи не було знайдено пару, варто взяти наступну за рейтингом підходящу групу та шукати пар групи не було знайдено пару, варто взяти наступну за рейтингом підходящу групу та шукати пару там. Процес повторюється, поки група, для мікросервісів якої шукали пари, стане порожньою, або ж поки не будуть проаналізовані всі доповнювальні групи.

Мікропроцесори, що не знайшли пари на цій ітерації, записують в окремий масив тих, які підуть на наступну ітерацію.

11. **Повторення ітерацій пошуку.** Якщо лишаються мікросервіси, які не знайшли доповнювачів, варто повторити кроки 7–10. Ітерації повторюють, поки утворюються пари. З кожною ітерацією кількість груп у пункті 7, на які діляться мікросервіси, зменшується пропорційно кількості екземплярів. Для вибору кількості кластерів є тільки емпіричні методи, такі як метод ліктя, силуетний аналіз, Gap-статистика та правило $\sqrt{n}$. Шаблони поведінки мікросервісів слід поділити на такі категорії для великої вибірки:

- робота в денний час: 9–17;
- робота в нічний час: 23–6;
- робота у вечірні години: 17–24;
- робота у ранкові години: 6–11;
- стабільне завантаження протягом доби;
- кілька варіацій завантаження власного типу (2–3).

Сформуємо 7–8 кластерів. Така кількість є резонною для великої вибірки. Якщо у групі лишатиметься менше ніж 2–3 елементи, нема сенсу створювати групу для такої малої кількості. Отже, пропонується користуватись правилом $\sqrt{n}$ у вибірці, що менша 8^2, а в більшій вибірці брати завжди 8 або 7 за умови регіонального розподілу мікросервісів.

ISSN 2788-6603

Останнє важливо, оскільки в інших країнах у разі зсуву часу робочі години можуть бути інші, тому аби коректніше знаходити доповнення, можна виділити більше груп кластеризації.

12. **Доповнення неможливі.** Може трапитися, що не всі мікросервіси знайшли доповнювальні навіть після 11-го кроку. Це може статись у таких випадках:

1) характер роботи мікросервісу випадковий. В такому разі йому варто виділити окремий сервер і не намагатись його сумістити з іншими;

2) екстремуми роботи мікросервісу дуже різкі, або ж у нестандартний час. В такому випадку пропонується поділити цей мікросервіс на кілька, якщо виражених екстремумів не більше двох, що описано в наступному пункті. В іншому разі – виділити окремий сервер під цей мікросервіс;

3) мікросервіси замалі, тому сумарне доповнення з іншими не дає повної завантаженості. Рішення – об'єднувати більш як два мікросервіси.

Підсумовуючи всі розв'язки вищезгаданих випадків, бачимо, що для групи мікросервісів будуть виділені сервери, а інші випадки слід звести до стандартних мікросервісів, які можуть знайти собі доповнення. Також, зважаючи на малий розмір і час застосування мікросервісів, що залишились, варто користуватись іншим алгоритмом їхнього збору (пункт 15).

13. **Розділення базового навантаження й екстремумів для одиноких особин.** Розглянемо випадок 12.2, коли існують мікросервіси з не дуже великим навантаженням, меншим за $P_{boundary} - P_{boundary} \times \alpha_{boundary}$, але все ж залишились без пари. Вірогідно, екстремуми трапляються в нестандартний час, або ж кількість денних завантажень переважає нічні завантаження, тому пари не існує. Проте є вірогідність знайти пару, якщо "згладити" піки роботи такого мікросервісу, коли їх небагато.

Для визначення стандартного навантаження та піків часового ряду можна проаналізувати значення середнього та стандартного відхилень (Zieffler, & Catalysts for Change, 2019), що допоможе виявити екстремуми. Знайдемо середнє значення часового ряду:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

Визначимо стандартне відхилення:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}.$$

Екстремуми визначають як значення, що перевищують певну кількість стандартних відхилень від середнього. В цій задачі одного стандартного відхилення вже достатньо для такої оцінки, тобто, якщо значення виходять за межі $\mu \pm \sigma$, їх можна вважати екстремумами (аномаліями). Нас цікавлять тільки пікові значення, бо їх можна "зрізати", виділивши під них окремий мікросервіс. Для монолітів указана практика схожа: доведеться створювати дублювальний моноліт у вибраний час і виконувати розподіл навантаження (load balancing) між двома екземплярами.

Має сенс зрізання всіх значень, які йдуть підряд і перевищують середнє. Для наочності наведено приклад. На рис. 3 показано часовий ряд із вираженим піком, через який важко знайти доповнювальний мікросервіс. Цей пік потрібно зрізати, тоді графік буде рівніший, знайти доповнення буде ймовірніше.

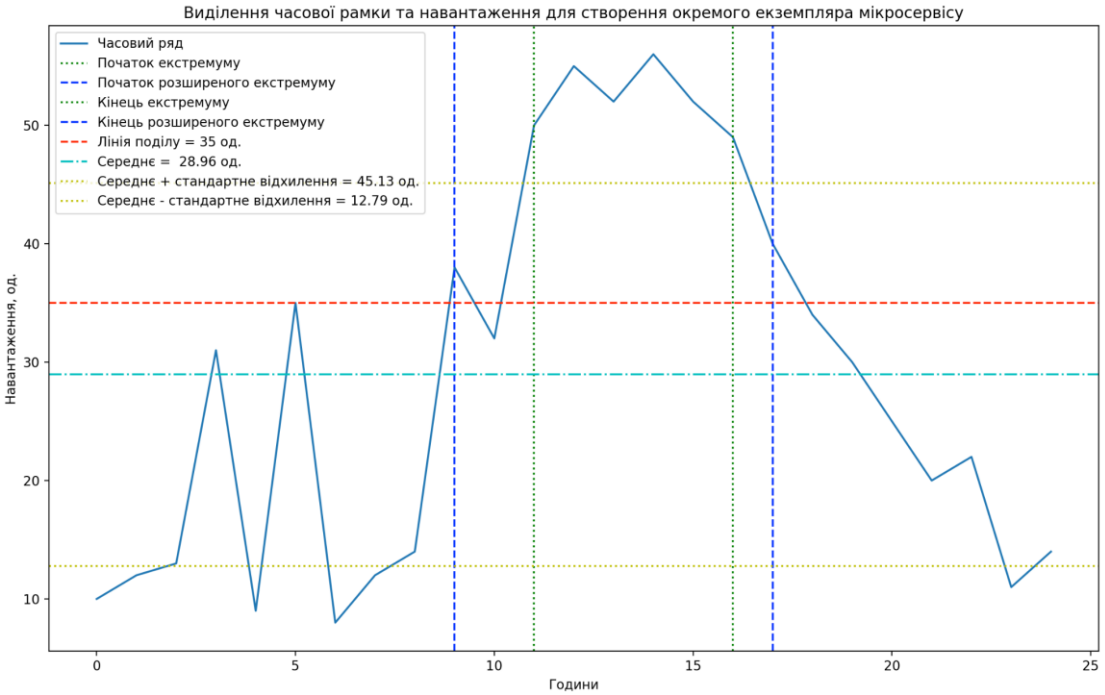


Рис. 3. Виділення навантаження мікропроцесора, яке варто запустити на іншому сервері. Стосовно моноліта – визначення часу для його додаткового запуску

За згаданим вище алгоритмом знайдуться пікові точки, що лежать вище верхньої жовтої лінії (з крапочок). Ці пікові значення позначено зеленими вертикальними лініями (з крапочок). Екстремуми, розміщені нижче нижньої жовтої лінії не розглядають. Купол графіка спостерігається довший час, ніж у верхньому квадраті, оточеному зеленими та жовтою лініями (з крапочок). Є сенс розширити цей проміжок у випадку, якщо він розташований вище середнього навантаження, позначеного голубою лінією (як крапка-тире). Для цього алгоритм проходить по всіх ближніх точках ліворуч і праворуч від піка, а також підбирає ті, що більші за середнє значення.

Відтак проводять пошук максимальної точки у часовому ряді, який не включає пік і близькі точки. Ця точка і буде навантаженням поділу мікросервісу. Наступним кроком буде відкидання всіх потенційно-пікових точок, що лежать між жовтою та голубою лініями, які менші за обраний новий максимум. Останній позначено червоною горизонтальною лінією (як тире). Розширений екстремум позначено синіми вертикальними лініями (тире).

На рис. 4 показано поділ мікросервісу на дві частини: до нової максимальної лінії та вище за неї.

Плануючи дії клауд-провайдера, має сенс заздалегідь запустити пікове навантаження на окремому сервері чи серверній групі, ввівши його в стан гарячого очікування (hot stand-by) (Levitin, Xing, & Dai, 2014).

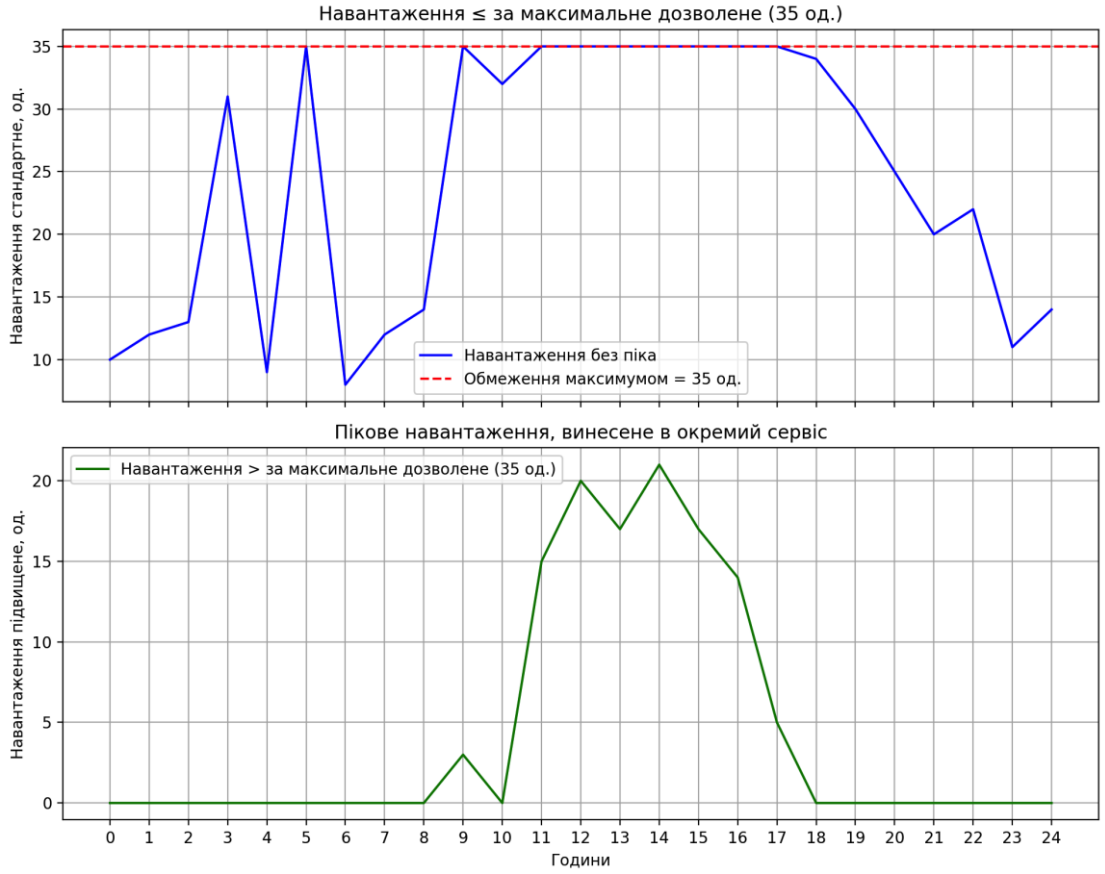


Рис. 4. Поділ мікросервісу на дві частини, екстремум виділено в окремий екземпляр

14. **Пошук доповнень для розділених мікропроцесорів**. Оскільки мікросервіси змінились із часу останнього пошуку доповнень, потрібно повторити крок 11.

15. **Комбінація малогабаритних мікросервісів.** Для решти мікросервісів є два наступні кроки, перший можна опустити:

- оскільки розмір мікросервісів малий, можна кластеризувати всі мікросервіси через KMeans, об'єднати по 2–3 мікросервіси з груп неперетину, у яких різні доповнювальні групи, після чого прогнати нові групи по кроку 11;
- решту мікросервісів, які залишаться, об'єднати, користуючись алгоритмом упакування множинного рюкзака із гнучким обмеженням цільової ваги (Cacchiani, Iori, Locatelli, & Martello, 2022).

Отже, всі мікропроцесори будуть згруповані, щоб обчислювальні потужності максимально не простоювали, а також щоб можна було підготувати потрібну машину для запуску обчислень на ній аби зменшити час розігріву, ввівши її в стан гарячого очікування (Levitin, Xing, & Dai, 2014).

16. **Налаштування під час роботи ПЗ на боці провайдера.** Після запуску система потребує продовження моніторингу. На цій стадії потрібно слідкувати за оновленням метрик і за якістю розподілу навантажень. Якщо виявлено, що протягом зазначеного часу серверна група використовує замалу кількість ресурсів, чи навпаки, потребує більше, то такі групи варто виокремити, скомбінувати з новими мікросервісами за нагоди й виконати пошук нових доповнень.

Результати

Під час алгоритму пошуку доповнювальних мікросервісів виконується кластеризація та екземпляри мікросервісів групуються у класи еквівалентності на основі подібності шаблона роботи. В межах груп екземпляри сортуються за

ISSN 2788-6603

амплітудою використання ресурсів, після чого групи, які визначаються як доповнювальні, перетинаються з метою пошуку конкретної пари комплементарних мікросервісів. У результаті, екземпляри зі значними відмінностями в шаблонах (протилежні шаблони навантаження) комбінуються з іншими, що мають подібні амплітуди, але протилежні фази, для максимального використання ресурсів. Комбінації з екземплярами, що мають низьку амплітуду, також можуть бути доречними для заповнення "дірок". У доповнювальних групах, алгоритм шукає перший екземпляр мікросервісу, що відповідає умовам доповнення (жадібний алгоритм). У разі виникнення труднощів знайти доповнення в першій найліпшій групі, зберігається статистика комбінацій з усіма екземплярами. Пошук продовжується до знаходження доповнень усім мікросервісам групи або поки не буде перебрано всі доповнювальні групи. Таких ітерацій проводять стільки, скільки будуть знаходитись пари, але "залишки" після пошуку комбінацій є очікуваними. Мікросервіси, що не знайшли доповнень додатково поділяють із використанням розширеного пошуку екстремумів за допомогою пошуку середнього та стандартного відхилень для збільшення їхніх шансів знайти компліменти. Кластеризацію та пошук проганяють до останньої успішної знахідки. Коли лишається невелика кількість мікросервісів, які не знайшли пари, комбінування відбувається за допомогою задачі про множинний рюкзак.

Результати роботи алгоритму – це скорочення кількості необхідних серверів до 15–20 %. Крім того, цей алгоритм дає можливість сформувати очікувану поведінку та ввести у стан розігріву додаткові мікросервіси, щоб користувачі ПЗ не відчували затримок у відповідях при їх різкому зростанні.

Відсоток економії серверів може варіюватись залежно від співвідношення локації хмарної технології до локації використання програмного забезпечення, а також від характеру розв'язуваних задач і збалансованості навантаження на хмарі, зокрема у денний і нічний час. Порівняно з поточним станом справ на хмарних ресурсах та Kubernetes, за потреби починає розгортатись новий екземпляр мікросервісу в момент, коли поточні метрики повідомлять про це (Скулиш, & Дмитренко, 2024; Дмитренко, & Скулиш, 2024b). Подібну ситуацію показано на рис. 4. Економія проявлятиметься у час, коли ресурс не повністю заповнений, тобто в години 8–11, 16–18, адже сервер мало завантажений у цей час, але, користуючись цим алгоритмом, він буде завантажений близько до повного.

Енергоефективність, яка буде наслідком економії серверних ресурсів, є однією з цілей ООН (United Nations Development Programme, 2024). Природними методами перероблення електроенергії у формат для вживання важко забезпечити потреби людства. Зменшуючи потреби в електроенергії, ми наближаємо кращу екологічну ситуацію у світі.

Дискусія і висновки

Наведено детальний опис алгоритму знаходження комплементарних мікросервісів, які за комбінування утворюють загальне навантаження на групу серверів, близьке до максимального. Цього можна досягти за допомогою різних технік поділу та поєднання мікросервісів, використовуючи вхідну та поточну статистику навантаження на канал, процесор або оперативну пам'ять.

Алгоритм містить нормалізацію вхідних результатів, кластеризацію методом K-Means, визначення доповнювальних груп, і на їхній основі – комплементарних пар мікросервісів, пропонує механізм поділу недоповнених мікросервісів на менші методом середнього та стандартного відхилень, а також передбачає комбінацію малогабаритних мікросервісів із більше ніж одним доповненням, користуючись методом упаковування множинного рюкзака. З метою візуалізації також запропоновано скористатись алгоритмами Z-Scale та FastDTW.

Внесок авторів: Олександра Дмитренко – розроблення методів і методології дослідження, опис результатів і написання висновків, огляд літературних джерел, збір емпіричних даних і проведення емпіричних досліджень; Марія Скулиш – поради щодо розроблення методів і методології дослідження й опису результату.

Список використаних джерел

Дмитренко О., & Скулиш, М. (2024b). Методи збору інформації та реалізації алгоритму доповнювальних навантажень. *Системи управління, навігації та зв'язку. Збірник наукових праць,* 4(78), Article 78. https://doi.org/10.26906/SUNZ.2024.4.056

Скулиш, М. А., & Дмитренко О. А. (2024). Метод організації мікросервісів на серверних групах Kubernetes. *Вчені записки Таврійського Національного Університету Імені В. І. Вернадського. Серія "Технічні науки"*, 1(5), 291–297. https://doi.org/10.32782/2663-5941/2024.5.1/41

Aspire Systems. (2023, June 22). Microservices Architecture: The Foundation of Cloud-Native Applications. *Software Engineering*. https://blog.aspiresys.com/software-product-engineering/microservices-architecture-the-foundation-of-cloud-native-applications/

Cacchiani, V., Iori, M., Locatelli, A., & Martello, S. (2022). Knapsack problems – An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. *Computers & Operations Research*, 143, 105693. https://doi.org/10.1016/j.cor.2021.105693

Dmytrenko, O., & Skulysh, M. (2024a). Method of Grouping Complementary Microservices Using Fuzzy Lattice Theory. In E. Siemens (Eds). *International Conference on Applied Innovations in IT (ICAIIT)*, 12(1), 11–18. Anhalt University of Applied Sciences Bernburg / Koethen / Dessau. https://doi.org/10.25673/115636

Dmytrenko, O., Skulysh, M., & Globa, L. (2024). Microservice Complimentary Groups Determination Algorithm for the Effective Resource Usage. In I. Sinitsyn & P. Andon (Eds.), Proceedings of the 14th International Scientific and Practical Programming Conference (UkrPROG 2024) (Vol. 3806, pp. 180–201). CEUR. https://ceur-ws.org/Vol-3806/#S_55_Dmytrenko_Skulysh_Globa

Levitin, G., Xing, L., & Dai, Y. (2014). Cold vs. Hot standby mission operation cost minimization for 1-out-of-*N* systems. *European Journal of Operational Research*, 234(1), 155–162. https://doi.org/10.1016/j.ejor.2013.10.051

Ozdemir, S., & Susarla, D. (2018). Feature Engineering Made Easy. Packt Publishing. https://www.oreilly.com/library/view/feature-engineering-made/9781787287600/

Salvador, S., & Chan, P. (2004). Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis*, 11, 80. https://cs.fit.edu/~pkc/papers/tdm04.pdf

Sharma, Y. (2023, September 26). *Key Strategies for Implementing AWS Network Load Balancer (AWS Community Builders).* DEV Community. https://dev.to/aws-builders/key-strategies-for-implementing-aws-network-load-balancer-35fc

Zieffler, A., & Catalysts for Change. (2019). Statistical Thinking: A Simulation Approach to Modeling Uncertainty (UM STAT 216 edition) (4.2).). In National Science Foundation (Eds.). MN: Catalyst Press. https://bookdown.org/frederick_peck/textbook/the-mean-and-standard-deviation.html MN: Catalyst Press. https://bookdown.org/frederick_peck/textbook/the-mean-and-standard-deviation.html

United Nations Development Programme. (2024). *Цілі сталого розвитку*. UNDP. https://www.undp.org/uk/ukraine/tsili-staloho-rozvytku

References

Aspire Systems. (2023, June 22). Microservices Architecture: The Foundation of Cloud-Native Applications. *Software Engineering*. https://blog.aspiresys.com/software-product-engineering/microservices-architecture-the-foundation-of-cloud-native-applications/

Cacchiani, V., Iori, M., Locatelli, A., & Martello, S. (2022). Knapsack problems – An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems. *Computers & Operations Research*,143, 105693. https://doi.org/10.1016/j.cor.2021.105693

Dmytrenko, O., & Skulysh, M. (2024a). Method of Grouping Complementary Microservices Using Fuzzy Lattice Theory. In E. Siemens (Eds). *International Conference on Applied Innovations in IT (ICAIIT)*, 12(1), 11–18. Anhalt University of Applied Sciences Bernburg / Koethen / Dessau. https://doi.org/10.25673/115636

Dmytrenko, O., & Skulysh, M. (2024b). Methods of data collection and implementation of accomplishing loads algorithm. Control, Navigation and Communication Systems. Academic Journal, 4(78), Article 78. https://doi.org/10.26906/SUNZ.2024.4.056

Dmytrenko, O., Skulysh, M., & Globa, L. (2024). Microservice Complimentary Groups Determination Algorithm for the Effective Resource Usage. In I. Sinitsyn & P. Andon (Eds.), Proceedings of the 14th International Scientific and Practical Programming Conference (UkrPROG 2024) (Vol. 3806, pp. 180–201). CEUR. https://ceur-ws.org/Vol-3806/#S_55_Dmytrenko_Skulysh_Globa

Levitin, G., Xing, L., & Dai, Y. (2014). Cold vs. Hot standby mission operation cost minimization for 1-out-of-N systems. European Journal of Operational Research, 234(1), 155–162. https://doi.org/10.1016/j.ejor.2013.10.051

Ozdemir, S., & Susarla, D. (2018). Feature Engineering Made Easy. Packt Publishing. https://www.oreilly.com/library/view/feature-engineering-made/9781787287600/

Salvador, S., & Chan, P. (2004). Toward Accurate Dynamic Time Warping in Linear Time and Space. In Intelligent Data Analysis (Vol. 11, p. 80). https://cs.fit.edu/~pkc/papers/tdm04.pdf

Sharma, Y. (2023, September 26). Key Strategies for Implementing AWS Network Load Balancer [AWS Community Builders]. DEV Community. https://dev.to/aws-builders/key-strategies-for-implementing-aws-network-load-balancer-35fc

Skulysh, M. A., & Dmytrenko, O. A. (2024). A Method of Organisation of Microservices on Kubernetes Server Groups. Scientific notes of Taurida National V. I. Vernadsky University. Series: Technical Sciences, 1(5), 291–297. https://doi.org/10.32782/2663-5941/2024.5.1/41

Zieffler, A., & Catalysts for Change. (2019). Statistical Thinking: A Simulation Approach to Modeling Uncertainty (UM STAT 216 edition) (4.2).). In National Science Foundation (Eds.). MN: Catalyst Press. https://bookdown.org/frederick_peck/textbook/the-mean-and-standard-deviation.html MN: Catalyst Press. https://bookdown.org/frederick_peck/textbook/the-mean-and-standard-deviation.html

United Nations Development Programme. (2024). Goals of sustainable development. UNDP. https://www.undp.org/uk/ukraine/tsili-staloho-rozvytku

Отримано редакцією журналу / Received: 21.10.24
Прорецензовано / Revised: 15.11.24
Схвалено до друку / Accepted: 20.11.24

Oleksandra DMYTRENKO, PhD Student, Assist.
ORCID ID: 0009-0009-4785-4226
e-mail: o_dmytrenko@lll.kpi.ua
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

Mariia SKULYSH, DSc (Engin.), Prof.
ORCID ID: 0000-0002-5141-1382
e-mail: mskulysh@gmail.com
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

MICROSERVICE GROUPING FOR UNDERUSED RESOURCE USAGE

Background. ***The task of optimizing cloud system resource usage is of high priority, as this technology has become widely adopted and easily accessible. Efficient services are common to be created to beastly adapt to different types of products, e.g. serverless technologies and proprietary databases, and cloud-optimized software. Performance testing of cloud applications has become significantly easier with the emergence of Docker and Kubernetes, which has further increased the demand for cloud resources. Since a considerable amount of resources is spent on the development and maintenance of cloud infrastructure, it is essential to understand how these resources can be optimized.***

Methods. ***The basic idea is to reorganize cloud infrastructure so that the software is distributed into the server groups, not the servers in a complementary manner, meaning that the hardware load will be close to the full load. To form complementary groups, methods of clusterization (K-Means), multiple knack's problem, sorting, binary search, shuffling and division based on mean and standard deviation are used. Additionally, FastDTW and Z-scale algorithms are proposed for visualization.***

Results. ***The article examines the key characteristics of computer systems that can be measured and influenced. These characteristics include transmission channel bandwidth, delay-based performance evaluation, the amount of RAM and permanent storage, processing power, and the number of cores. An algorithm is proposed to identify complementary microservice instances that could efficiently utilize server resources. Initially, microservice instances are classified by their capacity, considering small instances as a unit of capacity. Through analysis, the microservice instances are grouped into equivalence classes based on similarity. The instances are then sorted by the amplitude of resource usage. Ideally, instances with significant differences in load are combined with others that have similar amplitudes but opposite phases to maximize resource utilization. Combinations with instances that have low amplitude may also be appropriate. Within opposite classes, which differ in the activity phases of microprocessor instances, the algorithm searches for the first microservice instance that meets the conditions. To achieve this, statistics of combinations with all instances are stored until the first successful combination is found. Otherwise, the search continues till the least possible combination. The leftovers after the combinations search are expected. They are additionally divided using the extended extremums search with the mean and standard deviation search to increase their chances to find compliments. Finally, the small amount of microservices that didn't find a match is combined using multiple knack's problem.***

Conclusions. ***The conclusions of the article suggest an algorithm that will optimize the use of cloud system resources by effectively distributing the load between different microservices. Optimization means to maximize the use of server resources by filling its free time with other microservice and so reduce server downtime. This, in turn, will contribute to improving general cloud productivity and reducing maintenance costs of cloud infrastructures. This algorithm can be useful for cloud service providers and organizations that use cloud environments to deploy their applications.***

Keywords: ***microservice grouping, CPU efficiency, energy consumption optimization, cloud systems, microservice architecture, technology cost reduction.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

МЕРЕЖНІ
Й ІНТЕРНЕТ-ТЕХНОЛОГІЇ

УДК 004.7004.8
DOI: https://doi.org/10.17721/AIT.2024.1.09

Денис БОРОДАЙ, асп.
ORCID ID: 0009-0009-2531-056X
e-mail: agved2@gmail.com
Київський національний університет імені Тараса Шевченка, Київ, Україна

Юрій КРАВЧЕНКО, д-р техн. наук, проф.
ORCID ID: 0000-0002-0281-4396
e-mail: yurii.kravchenko@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

КОНЦЕПТУАЛЬНА МОДЕЛЬ ІНТЕЛЕКТУАЛІЗОВАНОГО КЕРУВАННЯ МЕРЕЖНИМИ РЕСУРСАМИ ЗА РЕАЛІЗАЦІЇ ПАРАДИГМИ SDN/NFV

Вступ. ***З розвитком і поширенням мережних технологій, проблема ефективного керування мережами зв'язку набуває все більшої актуальності. Одним із підходів для розв'язання цієї проблеми є використання віртуалізації мережних пристроїв, а саме концепції програмно-конфігурованих мереж (SDN) і віртуалізації мережних функцій (NFV), які дозволяють використовувати неспеціалізоване обладнання, включно з обладнанням на архітектурі процесорів x86, та роблять мережі ефективнішими за рахунок оптимального використання та розподілення ресурсів. Завдяки цьому з'являється можливість використання власних моделей і методів для керування мережами зв'язку. Метою роботи є пошук та розроблення концептуальної моделі інтелектуалізованого керування мережними ресурсами у програмно-конфігурованих мережах, яка на відміну від існуючих може динамічно виділяти ресурси, залежно від потреб мережі та бути пристосованою до можливих незначних розбіжностей між отриманими даними та реальними потребами в умовах дії зовнішніх і внутрішніх дестабілізуючих факторів.***

Методи. ***Використано нейромережні методи, методи нечіткої логіки й методи імітаційного моделювання.***

Результати. ***У роботі запропоновано концептуальну модель керування мережними ресурсами, що базується на гібридній нейронній мережі з використанням нечіткої логіки та нечіткого продукційного виведення (ANFIS), що дозволяє динамічно реагувати на потреби мережі та менш чутлива до швидкого старіння одержаних даних щодо її потреб. Отримана модель була протестована на змодельованій мережі зв'язку у середовищі MATLAB у двох сценаріях, з низьким і високим навантаженням на мережу, та показала хороший результат, який дозволяв віртуальним пристроям обробляти запити без перевантажень і мати певний запас обчислювальних можливостей для ефективної роботи й можливого незначного зростання потреб мережі.***

Висновки. ***Розроблена модель керування мережними ресурсами в середовищі MATLAB показала високу ефективність, що доводить необхідність та актуальність подальших досліджень із використання нечіткої логіки у керуванні мережами.***

Ключові слова: ***телекомунікаційна мережа, керування мережними ресурсами, нечітка логіка, нейронні мережі, віртуалізація мережних функцій, програмно-конфігуровані мережі, ANFIS, MATLAB.***

Вступ

З розвитком і поширенням інформаційно-комунікаційних технологій, обчислювальні пристрої та мережі передачі даних стали невід'ємною частиною будь-якої сфери. Інтернет є найважливішою інформаційною технологією сучасного життя, від функціонування якої залежать усі процеси й відносини в суспільстві (Плющ, Кравченко, & Труш, 2023). Чисельність та розміри центрів оброблення даних постійно зростають, а їхні мережі розширюються з урахуванням збільшення обсягів трафіка, профіль трафіка у цих мережах також постійно змінюється. Отже, конфігурація мережі повинна вміти адаптуватись із відповідною швидкістю. Для досягнення цього використовують віртуалізацію мережних пристроїв. Виділяють дві моделі мережної віртуалізації – програмно-конфігуровані мережі (SDN) і віртуалізацію мережних функцій (NFV). SDN – це мережа зв'язку, в якій керування мережею відокремлено від апаратних пристроїв передачі даних і реалізується програмно. NFV – це концепція, в якій мережні функції відокремлено від апаратних пристроїв. Використання зазначених технологій дозволяє створювати та керувати мережами не тільки на основі спеціалізованого мережного обладнання, а й на обчислювальних пристроях з архітектурою процесорів x86. Крім цього, це дозволяє спростити та здешевити підтримку мереж зв'язку, оптимізувати та збільшити їхню енергоефективність, а також зменшити кількість службового трафіка в них (Палагін, Євтушенко, & Гожий, 2021). Актуальність цих підходів і технологій підтверджує велика кількість наукових робіт. У роботах (Палагін, Євтушенко, & Гожий, 2021; Carrascal et al., 2023) розглянуто їхню актуальність, проблеми та можливості. У дослідженні (Суліма, & Скулиш, 2017) наведено гібридну систему керування ресурсами у мережах, які використовують концепції SDN і NFV. У роботі (Srinivas et al., 2021) розглянуто балансування навантаження в SDN-мережі з використанням машинного навчання.

З огляду на зазначене вище, дослідження та розроблення з використанням концепцій SDN і NFV є актуальними як з наукового, так і з комерційного погляду. Метою роботи постає пошук і розроблення концептуальної моделі інтелектуалізованого керування мережними ресурсами у програмно-конфігурованих мережах, яка на відміну від існуючих може динамічно виділяти ресурси, залежно від потреб мережі, та бути пристосованою до можливих незначних розбіжностей між отриманими даними та реальними потребами в умовах дії зовнішніх і внутрішніх дестабілізуючих факторів.

Постановка задачі. Відомо, що технології SDN і NFV дозволяють створювати оптимальніші й енергоефективніші мережі зв'язку. Це досягають правильним розподіленням обчислювальних можливостей між віртуальними вузлами зв'язку. Для досягнення цього, кожному віртуальному пристрою надають достатню кількість ресурсів, аби він міг обробити необхідну кількість запитів без перевантажень, але не занадто велику, аби не було простою. Отже, в основі оптимального керування мережними ресурсами лежить оптимізаційна задача з пошуку оптимальної кількості обчислювальних спроможностей, які необхідно виділити для кожного віртуального мережного пристрою. У деяких розглянутих роботах (Суліма, & Скулиш, 2017; Скулиш, & Суліма, 2019) як вхідні дані для керування такими мережами,

ISSN 2788-6603

крім актуальних даних, подають також прогнозовані дані. Проте навіть такий підхід мав свої недоліки. Телекомунікаційні навантаження формують відповідні довготермінові зміни та короткотермінові коливання параметрів функціонування мережі. Але довготермінові коливання можуть бути передбачені з високої достовірністю, шляхом аналізу змін у минулому. Варто підкреслити те, що короткотермінові зміни менш передбачувані або зовсім не передбачувані (Скулиш, & Суліма, 2019). Крім цього, дані отримані у реальному часі, можуть швидко втрачати актуальність і тим самим не відображати реальні потреби мережі.

Отже, постає задача пошуку та розроблення концептуальної моделі інтелектуалізованого керування мережними ресурсами у програмно-конфігурованих мережах, яка на відміну від існуючих може динамічно виділяти ресурси, залежно від потреб мережі та бути пристосованою до можливих незначних розбіжностей між отриманими даними та реальними потребами в умовах дії зовнішніх і внутрішніх дестабілізуючих факторів. Додатково, можна дослідити ефективність отриманих моделей на змодельованій мережі зв'язку.

Методи

Для розв'язання поставленої задачі в роботі пропонується використовувати нейромережні методи, методи нечіткої логіки й методи імітаційного моделювання. В основі концептуальної моделі керування мережними ресурсами лежатиме адаптивна система нейро-нечіткого виведення (adaptive neuro-fuzzy inference system, ANFIS) (Jang, 1993), яка поєднує принципи нейронних мереж і принципи нечіткої логіки, що дозволяє поєднувати переваги обох підходів. Завдяки використанню принципу нейронної мережі система може навчатися на основі отриманих даних і реагувати на зміни потреб мережі, а за рахунок використання системи нечіткої логіки, вхідні дані розглядатимуться як нечітке значення, що теоретично дозволить знизити чутливість моделі до незначних розбіжностей між отриманими даними та потребами мережі. На вхід моделі подаватимуться дані щодо кількості запитів від кінцевих пристроїв, а на виході будемо отримувати оцінку, щодо необхідної кількості обчислювальних можливостей для віртуального маршрутизатора. Для перевірки моделі моделюватимемо мережу зв'язку та перевірятимемо на ній отриману модель у середовищі MATLAB.

Результати

Для створення та тестування моделей використовуватимемо змодельовану мережу зв'язку, схему якої зображено на рис. 1. Для кожного віртуального маршрутизатора була створена модель, на основі ANFIS, на вхід якої подаватимуться дані X_n щодо кількості запитів, а на виході буде необхідна кількість обчислювальних можливостей Y_k, які потрібно виділити.

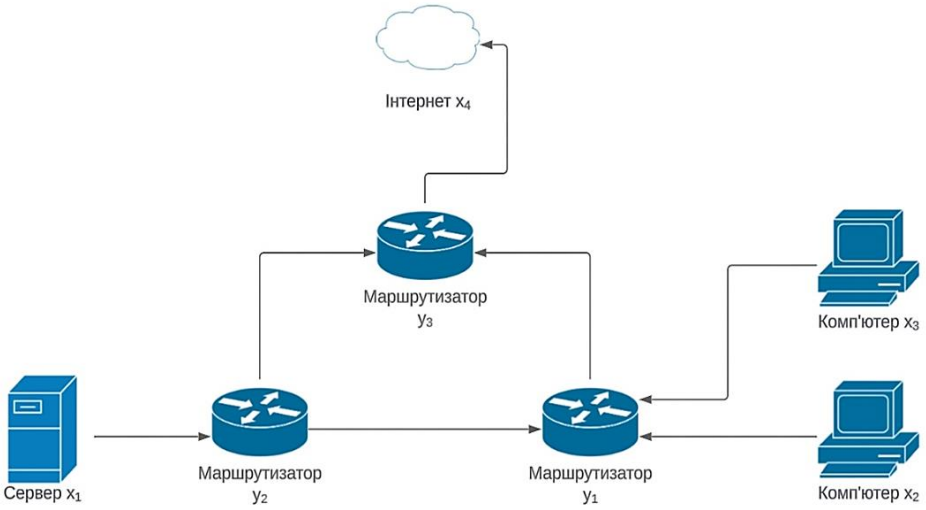


Рис. 1. Схема досліджуваної мережі зв'язку

У вказаній мережі, комп'ютери x_2 та x_3 можуть здійснювати запити до сервера x_1 і зовнішні запити в інтернеті x_4, або відповідати на аналогічні запити. Сервер x_1 може відповідати на запити, або здійснювати їх до комп'ютерів x_2, x_3 та інтернету x_4. З інтернету x_4 можуть відповідати на запити, або здійснювати їх до комп'ютерів x_2, x_3 та сервера x_1, або здійснювати аналогічні запити. Для формалізації та спрощення наведених умов наведемо коротший запис із поясненнями у табл. 1.

У межах симуляції вважатимемо, що кожна умова має одиниці виміру запит/с та має свою область визначення, а саме X_1, X_2, X_4, X_5, X_6, X_7, X_9, X_{10} = [10; 100] та X_3, X_8 = [100; 5000]. Як шуканий параметр, знаходитимемо необхідну кількість обчислювальних можливостей для маршрутизаторів y_1, y_2 та y_3, які будуть також вимірюватися у запит/с та позначатися Y_1, Y_2 та Y_3 відповідно.

Як уже сказано, моделі керування мережними ресурсами базуватимуться на нейро-нечіткій мережі ANFIS. Функцією приналежності буде функція приналежності Гаусса, яка визначається таким рівнянням:

$$f(x) = e^{\frac{-(x-c)^2}{2\sigma^2}},$$

де x – вхідне значення, σ – середнє квадратичне відхилення, c – середнє значення. Приклад графічного зображення функції зображено на рис. 2.

Таблиця 1

Спрощений запис запитів у мережі зв'язку

Спрощений запис	Суть запису
X_1	Сервер x_1 відповідає на запит комп'ютера x_2 або здійснює запит до комп'ютера x_2
X_2	Сервер x_1 відповідає на запит комп'ютера x_3 або здійснює запит до комп'ютера x_3
X_3	Сервер x_1 відповідає на зовнішній запит з інтернету x_4 або здійснює зовнішній запит в інтернеті x_4
X_4	Комп'ютер x_2 здійснює запит у сервера x_1 або відповідає на запит від сервера x_1
X_5	Комп'ютер x_2 здійснює зовнішній запит в інтернеті x_4 або відповідає на зовнішній запит в інтернеті x_4
X_6	Комп'ютер x_3 здійснює запит у сервера x_1 або відповідає на запит від сервера x_1
X_7	Комп'ютер x_3 здійснює зовнішній запит в інтернеті x_4 або відповідає на зовнішній запит в інтернеті x_4
X_8	З інтернету x_4 відповідають на запит сервера x_1 або здійснюють запит до сервера x_1
X_9	З інтернету x_4 відповідають на запит комп'ютера x_2 або здійснюють запит до комп'ютера x_2
X_{10}	З інтернету x_4 відповідають на запит комп'ютера x_3 або здійснюють запит до комп'ютера x_3

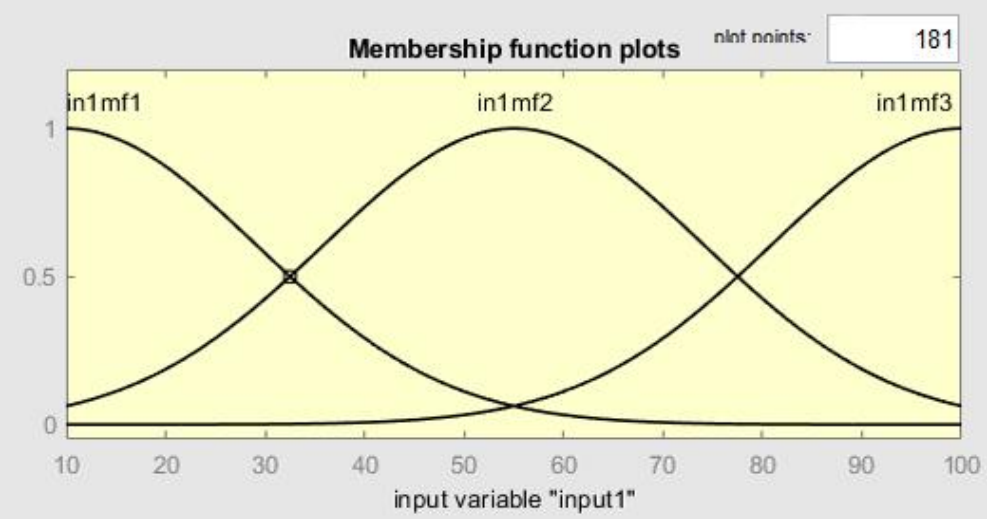


Рис. 2. Графічне зображення функцій приналежності Гаусса для X_1

Неважко помітити, що кожен маршрутизатор не оброблюватиме всі запити. Наприклад, маршрутизатор y_3 не буде оброблювати внутрішні запити, такі як X_1, X_2, X_4 та X_6. Враховуючи це, можна оптимізувати вхідні дані для кожної моделі, відкинувши непотрібні вхідні дані, тим самим знизивши їхню розмірність і кількість нечітких правил. Для Y_1 вхідні дані будуть мати такий вигляд: $[X_1, X_2, X_4, X_5, X_6, X_7, X_9, X_{10}]$, для Y_2 $[X_1, X_2, X_3, X_4, X_6, X_8]$, а для Y_3 $[X_3, X_5, X_7, X_8, X_9, X_{10}]$.

Продукційні правила у ANFIS доцільно задавати, використовуючи кон'юнкцію. Наприклад, покажемо одне з нечітких правил для визначення Y_2, яке виглядатиме так:

$$< \text{ЯКЩО } (X_1 \in In_1Mf_1) \wedge (X_2 \in In_2Mf_1) \wedge (X_3 \in In_3Mf_1) \wedge (X_4 \in In_4Mf_1) \wedge$$

$$\wedge (X_6 \in In_6Mf_1) \wedge (X_8 \in In_8Mf_1), \quad \text{ТО } Y_2 \in OutMf_1 >,$$

де In_iMf_j – функція приналежності j для входу i, $OutMf_k$ – нечіткий вивід k.

Для дослідження ефективності отриманих моделей протестуємо їх у змодельованій моделі зв'язку в декількох сценаріях. Як критерії використовуватимемо такі правила: якщо модель виділила недостатню кількість обчислювальних ресурсів – то її роботу оцінюємо незадовільно; якщо модель виділила достатню кількість ресурсів, але не більше 15 % необхідного – то модель добре впоралась із поставленою задачею; якщо модель виділила більше 15 % необхідної кількості ресурсів, але менше 25 % – то модель оцінюємо задовільно; якщо ж модель виділила більше 25 % необхідного – модель виконала роботу незадовільно.

ISSN 2788-6603

Для початку розглянемо варіант із незначним навантаженням на мережу, вхідні дані будуть виглядати так: $X_1 = 25$, $X_2 = 15$, $X_3 = 500$, $X_4 = 30$, $X_5 = 10$, $X_6 = 10$, $X_7 = 20$, $X_8 = 700$, $X_9 = 15$, $X_{10} = 35$. У цьому випадку комп'ютер x_2 робить невелику кількість запитів до сервера, та майже не здійснює зовнішніх запитів у інтернет, комп'ютер x_3 навпаки майже не робить запитів до сервера, але здійснює невелику кількість зовнішніх запитів до інтернету, сервер x_1 та інтернет x_4 здійснюють певну кількість взаємних запитів і відповідей один одному, та відносно пропорційну кількість до комп'ютерів x_2 та x_3. Беручи до уваги ці дані, можна вирахувати, що мінімальна кількість необхідних обчислювальних можливостей для віртуального маршрутизатора y_1 становить 160 запитів/с, для y_2 – 1280 запитів/с, а для y_3 також 1280 запитів/с. Результат моделі для цього сценарію зображено на рис. 3.

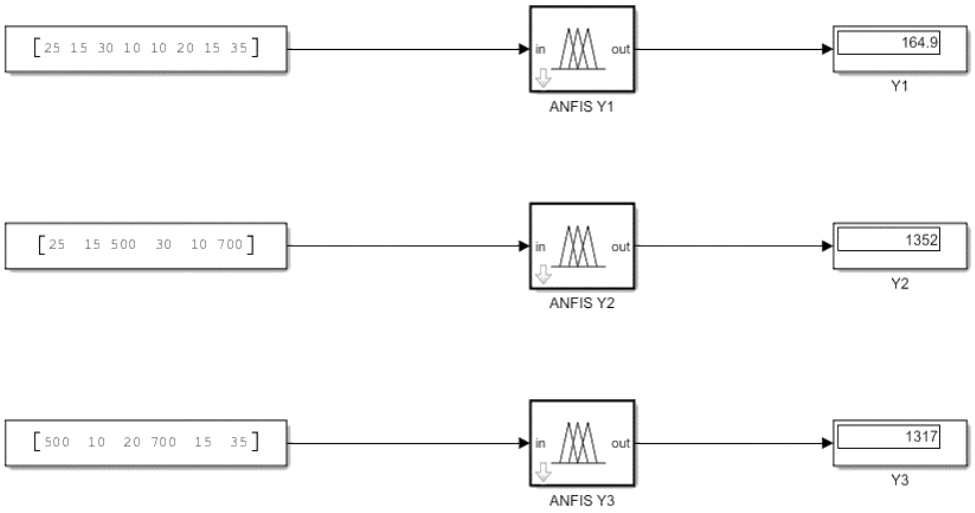


Рис. 3. Результат роботи моделі для сценарію з незначним навантаженням

Отже, моделі розрахували такі параметри: $Y_1 = 165$, $Y_2 = 1352$, $Y_3 = 1317$. У всіх випадках віртуальні маршрутизатори зможуть обробити запити без перевантажень і затримок, а також мають деяку кількість додаткової обчислювальної можливості у межах від 2,9 % до 5,6 % необхідної, що можна вважати хорошим результатом.

Розглянемо сценарій зі значним навантаженням на мережу, вхідні дані виглядатимуть як $X_1 = 100$, $X_2 = 50$, $X_3 = 4500$, $X_4 = 80$, $X_5 = 30$, $X_6 = 45$, $X_7 = 100$, $X_8 = 3000$, $X_9 = 75$, $X_{10} = 100$. У цьому випадку комп'ютер x_2 робить велику кількість запитів до сервера та певну кількість зовнішніх запитів у інтернет, комп'ютер x_3 робить певну кількість запитів до сервера та здійснює велику кількість зовнішніх запитів до інтернету, сервер x_1 та інтернет x_4 здійснюють велику кількість взаємних запитів і відповідей один одному, та відносно пропорційну кількість до комп'ютерів x_2 та x_3. З огляду на ці дані, можна вирахувати, що мінімальна кількість необхідних обчислювальних можливостей для віртуального маршрутизатора y_1 становить 580 запитів/с, для y_2 – 7775, а для y_3 також 7805 запитів/с. Результат моделі для такого сценарію представлено на рис. 4.

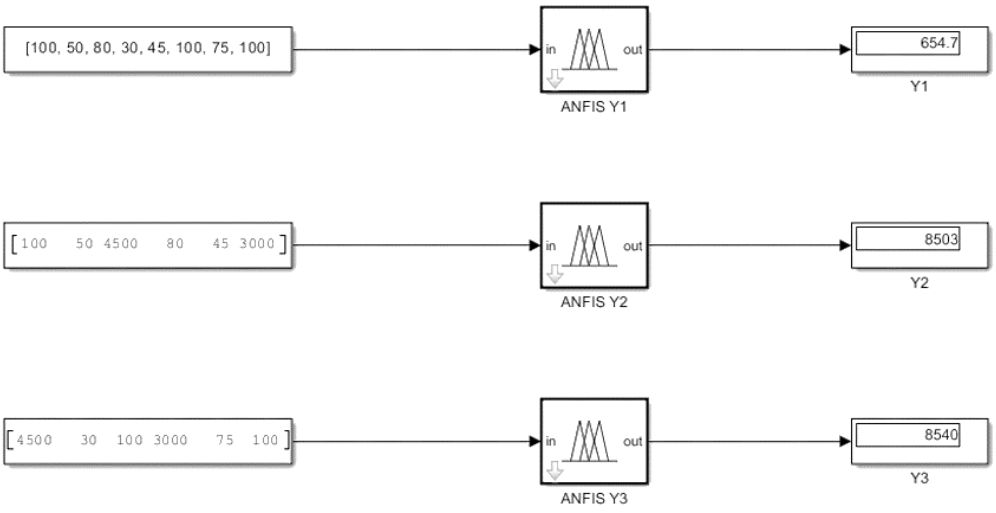


Рис. 4. Результат роботи моделі для сценарію зі значним навантаженням

За результатами моделювання отримано такі параметри: $Y_1 = 655$, $Y_2 = 8503$, $Y_3 = 8540$. Отже, в усіх випадках віртуальні маршрутизатори зможуть обробити запити без перевантажень і затримок, а також мають деяку кількість додаткової обчислювальної можливості, а саме на 9,4 % більше необхідної для маршрутизаторів y_2 та y_3, на які припадають значно більші навантаження, та 12,9 % для маршрутизатора y_1 із меншим навантаженням.

Дискусія і висновки

Створено концептуальну модель інтелектуалізованого керування мережними ресурсами, яка на відміну від існуючих ґрунтується на нечіткому продукційному виведенні за реалізації технологій віртуалізації мережних функцій для програмно-конфігурованих мереж. Запропонована модель реалізує стратегію незалежного масштабування ресурсів керування та передачі даних для забезпечення мережі переваги парадигми SDN / NFV. Додатково перевірено ефективність отриманих моделей, яка відбувалася на змодельованій мережі зв'язку у двох сценаріях. Отримані результати свідчать, що створені моделі добре впорались зі змодельованими сценаріями, із чого випливає, що моделі, які базуються на нечіткій логіці, мають потенціал для подальшого дослідження.

Внесок авторів: Юрій Кравченко – розроблення концептуальної моделі; Денис Бородай – огляд літературних джерел, збір емпіричних даних, проведення комп'ютерного моделювання, опис результатів і написання висновків.

Список використаних джерел

Палагін, В., Євтушенко, І., & Гожий, О. (2021). Віртуалізація як середовище реалізації мережевих функцій. *Вісник Черкаського державного технологічного університету*, 2, 31–38. https://doi.org/10.24025/2306-4412.2.2021.234703

Плющ, О., Кравченко, Ю., & Труш, О. (2023). Рекурентний алгоритм проєктування телекомунікаційних систем і мереж. *Сучасні інформаційні технології*, 1(2), 64–72. https://doi.org/10.17721/AIT.2023.1.10

Скулиш, М., & Суліма, С. (2019). Керування ресурсами для віртуалізованих мережевих функцій. У В. М. Безрука, Л.С. Глоби, О.Є Стрижак (Ред.). *Наукоємні технології оптимізації та керування в інфокомунікаційних мережах* (с. 97–126). Інститут обдарованої дитини НАПН України. https://ela.kpi.ua/items/3533990e-c8c9-464e-84c2-1c77516f7292

Суліма, С., & Скулиш, М. (2017). Гібридна система управління ресурсами для віртуалізованих мережевих функцій. *Радіоелектроніка, інформатика, управління*, 1, 16–23. https://doi.org/10.15588/1607-3274-2017-1-2

Carrascal, D., Rojas, E., Arco, J., Lopez-Pajares D., Alvarez-Horcajo, J., & Carral J. (2023). A Comprehensive Survey of In-Band Control in SDN: Challenges and Opportunities. *Electronics*, 12(6), 1265. https://doi.org/10.3390/electronics12061265

Jang, J. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665–685. https://doi.org/10.1109/21.256541

Srinivas, J., Sakthivel, S., Sudha, R., Rohit, K., Ranjan, W., & Lokesh, M. (2021). SDN network load balancing using environmental congenital ACO methodology. *International Journal of Biology, Pharmacy and Allied Sciences (IJBPAS),* 10(11), 913–923. http://dx.doi.org/10.31032/IJBPAS/2021/10.11.1079

References

Carrascal, D., Rojas, E., Arco, J., Lopez-Pajares D., Alvarez-Horcajo, J., & Carral J. (2023). A Comprehensive Survey of In-Band Control in SDN: Challenges and Opportunities. *Electronics*, 12(6), 1265; https://doi.org/10.3390/electronics12061265

Jang, J. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665–685. https://doi.org/10.1109/21.256541

Palahin, V., Yevtushenko, I., & Hozhyi, O. (2021). Virtualization as an environment of realization of network functions. *Bulletin of Cherkasy State Technological University*, (2), 31–38 [in Ukrainian]. https://doi.org/10.24025/2306-4412.2.2021.234703

Pliushch, O., Kravchenko, Y., & Trush, O. (2023). Recurrent algorithm of telecommunication systems and networks design. *Advanced Information Technology*, 1(2), 64–72 [in Ukrainian]. https://doi.org/10.17721/AIT.2023.1.10

Skulysh, M., & Sulima, S. (2019). Resource management for virtualized network functions. In V. Bezruka, L. Globa, O. Strizhak (Eds). *Science-intensive optimization and control technologies in information communication networks* (pp. 97–126). Institute of the gifted child of the National Academy of educational sciences of Ukraine [in Ukrainian]. https://ela.kpi.ua/items/3533990e-c8c9-464e-84c2-1c77516f7292

Srinivas, J., Sakthivel, S., Sudha, R., Rohit, K., Ranjan, W., & Lokesh, M. (2021). SDN network load balancing using environmental congenital ACO methodology. *International Journal of Biology, Pharmacy and Allied Sciences (IJBPAS),* 10(11), 913–923. http://dx.doi.org/10.31032/IJBPAS/2021/10.11.1079

Sulima, S., & Skulysh, M. (2017). Hybrid resource provisioning system for virtual network functions. *Radio Electronics, Computer Science, Control*, 1, 16–23 [in Ukrainian]. https://doi.org/10.15588/1607-3274-2017-1-2

Отримано редакцією журналу / Received: 15.08.24
Прорецензовано / Revised: 20.10.24
Схвалено до друку / Accepted: 27.10.24

ISSN 2788-6603

Denys BORODAI, PhD Student
ORCID ID: 0009-0009-2531-056X
e-mail: agved2@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

Yurii KRAVCHENKO, DSc (Engin.), Prof.
ORCID ID: 0000-0002-0281-4396
e-mail: yurii.kravchenko@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

THE CONCEPTUAL MODEL OF INTELLIGENT MANAGEMENT OF NETWORK RESOURCES IN THE IMPLEMENTATION OF THE SDN/NFV PARADIGM

Background. ***With the development and spread of network technologies, the problem of effective management of networks is becoming increasingly relevant. One approach to solving this problem is to use network virtualization, such as software-defined networking (SDN) and network functions virtualization (NFV), which allows to use of non-specialized hardware, including hardware on the x86 architecture, and makes networks more efficient due to the optimal use and distribution of resources. Due to this, it becomes possible to use own models and methods for managing networks. The purpose of the work is to find and develop a conceptual model of intelligent management of network resources in software-defined networks, which, unlike the existing ones, can dynamically allocate resources depending on the needs of the network and can adapt to possible minor discrepancies between the received data and real needs under the conditions of external and internal destabilizing factors.***

Methods. ***In this paper methods of neural networks, fuzzy logic methods as well as methods of computer simulation are used.***

Results. ***In this study, it is proposed the conceptual model of network resource management based on a hybrid neural network using fuzzy logic and fuzzy output inference (ANFIS), which allows for dynamic response to network needs and is less sensitive to the rapid obsolescence of received data regarding its needs. The model was tested on a simulated communication network in the MATLAB environment in two scenarios with low and high network load. It showed a good result that allowed the virtual devices to handle requests without overloads and have a certain margin of computing resources for efficient work and possible minor growth of network needs.***

Conclusions. ***The developed model of network resource management in the MATLAB environment showed high efficiency, which proves the necessity and relevance of further research on using fuzzy logic in network management.***

Keywords: ***communication network, network resource management, fuzzy logic, neural networks, network functions virtualization, software-defined networking, ANFIS, MATLAB.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

УДК 656.11:004.42
DOI: https://doi.org/10.17721/AIT.2024.1.10

Людмила ВОЛОЩУК, канд. техн. наук, доц.
ORCID ID: 0000-0002-2510-0038
e-mail: lavstumbre@gmail.com
Одеський національний університет імені І. І. Мечникова, Одеса, Україна

Олександр СБІТНЄВ, асп.
ORCID ID: 0009-0008-6311-612X
e-mail: alexsbitnev99@gmail.com
Одеський національний університет імені І. І. Мечникова, Одеса, Україна

ГІБРИДНА ХМАРНА ІТЕЛЕКТУАЛЬНА ТРАСПОРТНА IOT-СИСТЕМА МОНІТОРИНГУ ДОРОЖНЬОГО ТРАФІКА ЖИТЛОВОГО МІКРОРАЙОНУ

Вступ. ***Запропоновано нову архітектуру інтелектуальної транспортної системи (ІТС), що використовує технології інтернету речей (IoT) і хмарну платформу Azure. Наукова новизна полягає у розробленні архітектури, що поєднує граничні обчислення, хмарні сервіси й алгоритми машинного навчання для адаптивного керування дорожнім рухом на основі даних у реальному часі. Розроблена архітектура дозволяє ефективно обробляти інформацію про стан транспортних потоків, здійснювати їхнє моделювання й автоматично регулювати сигнали світлофорів із метою зниження заторів. Ефективність архітектури перевірено через серію експериментів, спрямованих на розпізнавання транспортних засобів, оптимізацію керування світлофорами та моніторинг транспортної ситуації в реальному часі.***

Методи. ***Використано метод імітаційного комп'ютерного моделювання для керування інтелектуальною транспортною системою, метод навчання з підкріпленням для навчання інтелектуальної транспортної системи, метод комп'ютерного зору для розпізнавання транспортних засобів.***

Результати. ***Запропонована архітектура ІТС базується на IoT-технологіях і дозволяє збирати й аналізувати дані про дорожній трафік у режимі реального часу. Розроблену систему протестовано на різних ділянках міського мікрорайону з різними рівнями транспортного навантаження. Експерименти показали, що система здатна адаптивно змінювати сигнали світлофорів на основі аналізу транспортної ситуації, що дозволяє значно покращити пропускну спроможність доріг і зменшити затори.***

Висновки. ***Результати проведених експериментів підтвердили ефективність запропонованої архітектури інтелектуальної транспортної системи. У подальших дослідженнях можливо вдосконалити систему завдяки впровадженню складніших алгоритмів штучного інтелекту для автоматизації прийняття рішень щодо керування світлофорами.***

Ключові слова: ***інтелектуальна транспортна система, хмарна платформа, оптимізація транспортної системи, проєктування архітектури IoT-системи.***

Вступ

Інтелектуальна транспортна система (ІТС) – це система керування транспортом, що використовується для прогнозування і керування транспортними потоками. Ця система орієнтована на моделювання різноманітних подій і прогнозування потенційно небезпечних ситуацій, забезпечуючи прийняття рішень в умовах високої складності й оброблення великих обсягів даних (Hunt, Robertson, & Bretherton,1981).

ІТС можна розглядати як ключовий елемент сучасного підходу до підвищення ефективності роботи транспортної системи країни, завдяки розширенню інформаційної інфраструктури: автоматизованому збору даних у реальному часі, моделюванню й оперативному впливу на керування транспортними потоками.

"Інтернет речей" (IoT) – це концепція, яка передбачає підключення до глобальної мережі будь-яких об'єктів навколишнього середовища. Це можуть бути промислові і побутові пристрої, здатні передавати корисну інформацію, запобігати несанкціонованому доступу й автоматично обмінюватися даними без втручання людини.

Для керування мережею IoT використовують хмарні платформи – спеціалізоване програмно-апаратне забезпечення, яке дозволяє підключати пристрої до хмарної інфраструктури та дистанційно ними керувати.

Інтернет речей і штучний інтелект (AI) відкривають можливості для створення нових розумних транспортних систем для наземного, повітряного, залізничного та морського транспорту. Ці рішення сприяють об'єднанню транспортних засобів, світлофорів, пунктів оплати й іншої інфраструктури, що дозволяє зменшити затори, запобігти аваріям, знизити викиди та підвищити ефективність транспортних процесів (Katerna, 2019).

Ефективне керування дорожнім рухом є критичним завданням в урбаністичному плануванні. Традиційні системи керування світлофорами часто спираються на фіксовані графіки, які не можуть адаптуватися до реальних умов дорожнього руху. У цій статті розглянуто новий підхід, що використовує IoT і хмарні обчислення для створення динамічної та адаптивної системи керування дорожнім рухом. Запропонована система інтегрує різні технології для моніторингу й оптимізації руху транспорту, що покращує загальну ефективність і безпеку дорожнього руху (El-Tantawy, Abdulhai, & Abdelgawad, 2014).

У цій статті запропоновано нову архітектуру системи моніторингу дорожнього трафіка, яка базується на технологіях інтернету речей і хмарній платформі Azure. Наукова новизна роботи полягає в інтеграції граничних обчислень, хмарних сервісів і алгоритмів машинного навчання для адаптивного керування дорожнім рухом на основі даних у реальному часі.

Постановка задачі. Задачею статті є розроблення та дослідження моделі інтелектуальної транспортної системи, яка використовує технології інтернету речей і хмарну платформу Azure для адаптивного керування світлофорним регулюванням на основі реальних даних про транспортні потоки. До основних задач дослідження належать такі.

1) Розроблення системи розпізнавання транспортних засобів за допомогою моделі YOLOv3 та бібліотеки OpenCV для отримання даних про типи й кількість транспортних засобів на перехрестях.

ISSN 2788-6603

2) Моделювання дорожньої ситуації та оптимізація часу сигналів світлофорів за допомогою алгоритму навчання з підкріпленням Q-Learning.

3) Інтеграція IoT і хмарних технологій для моніторингу й керування транспортною системою через панель керування на платформі Azure IoT Central.

4) Оцінювання ефективності запропонованої системи на основі проведеного моделювання та тестування в умовах, наближених до реальних.

Це дослідження має на меті створення комплексного рішення, яке поєднує технології комп'ютерного зору, машинного навчання та хмарного оброблення даних для покращення керування дорожнім рухом в умовах сучасного міста.

Методи

Для розроблення ефективної інтелектуальної транспортної системи вивчено багато робіт, які досліджують використання IoT-технологій і хмарних платформ для керування транспортними потоками. Зокрема і робота Nadeem Abbas та інших зосереджена на використанні хмарних обчислень для оброблення великих обсягів даних трафіка й інтеграції туманних обчислень для більш локалізованого керування. Ця робота послужила основою для розроблення архітектури системи, яка використовує як хмарні, так і туманні обчислення для масштабованості та швидкості реагування на трафік. Однак ця робота робить значний внесок у частині оптимізації керування світлофорним регулюванням, використовуючи адаптивні алгоритми машинного навчання для реального часу, що не розглядалося так глибоко в роботі Abbas.

Іншою важливою основою є дослідження (Gayratov, Kilichov, & Toshpulatov, 2022), де запропоновано багаторівневу архітектуру для розподілених IoT-систем, що дозволяє об'єднати різні рівні керування даними і контролем. Це дослідження також використовує багаторівневу архітектуру, проте воно розширене додаванням рівня адаптивного регулювання світлофорів, що базується на машинному навчанні, зокрема і на методі Q-Learning, який значно підвищує ефективність керування в умовах змінного трафіка.

У роботі проаналізовано існуючі інтелектуальні транспортні системи, зокрема і в дослідженнях (Клюєва, Цимбала, & Сігоніна, 2023) також розглянуто історію їхнього розвитку. Це дозволило визначити переваги та недоліки сучасних систем.

Запропонована система вирізняється використанням багаторівневої архітектури, яка поєднує граничні обчислення (Azure Stack Edge) із хмарним обробленням, що забезпечує ефективне керування трафіком у реальному часі. Основні нововведення полягають у локальному обробленні даних і адаптивному керуванні світлофорами на основі аналізу транспортних потоків.

У дослідженні використано кілька ключових методів, кожен з яких відіграє важливу роль у розробленні та впровадженні інтелектуальної транспортної системи (Сбітнєв, 2022).

Метод імітаційного комп'ютерного моделювання. Імітаційне комп'ютерне моделювання є підходом, який дозволяє відтворювати й аналізувати поведінку складних систем в умовах реального часу або за експериментальних умов. Цей метод базується на використанні математичних моделей для симуляції процесів або систем, що дозволяє дослідникам передбачати результати різних сценаріїв без необхідності впровадження їх у реальному середовищі. Імітаційне моделювання часто використовують у випадках, коли експерименти в реальному світі є занадто дорогими, небезпечними або трудомісткими (Danshyna, Nechausov, & Andrieiev, 2022).

У цій роботі метод імітаційного комп'ютерного моделювання використано для керування інтелектуальною транспортною системою, що дозволило моделювати різні сценарії дорожньої ситуації та оцінювати ефективність запропонованих рішень з оптимізації світлофорного регулювання.

Метод навчання з підкріпленням (Reinforcement Learning). Навчання з підкріпленням є підходом у машинному навчанні, що базується на концепції агента, який взаємодіє із середовищем і отримує зворотний зв'язок у вигляді винагороди або покарання залежно від виконаних дій. Агент намагається максимізувати сукупну винагороду, навчаючись на основі досвіду, що дозволяє знаходити оптимальні стратегії для досягнення поставлених цілей.

У цій роботі метод навчання з підкріпленням використано для навчання інтелектуальної транспортної системи з метою адаптивного керування сигналами світлофорів. Це дозволило системі самостійно навчатися на основі даних у реальному часі, що сприяло оптимізації дорожнього руху.

Метод комп'ютерного зору (Computer Vision). Комп'ютерний зір є галуззю штучного інтелекту, яка дозволяє комп'ютерам розпізнавати й інтерпретувати візуальну інформацію з реального світу. Цей метод базується на використанні алгоритмів, що аналізують зображення або відео для ідентифікації об'єктів, виявлення певних характеристик або проведення інших видів аналізу.

Для навчання моделі вибрано алгоритм навчання з підкріпленням (Q-Learning). Алгоритм Q-Learning використовують для оптимізації регулювання світлофорів за допомогою навчання системи на основі досвіду взаємодії з дорожнім середовищем. Основні компоненти алгоритму включають стан (s), що відображає поточну дорожню ситуацію, дію (a), яку агент може здійснити (наприклад, змінити сигнал світлофора), та нагороду (r), що оцінює якість рішення на основі таких показників, як час очікування або довжина черги (Wiering, & Otterlo, 2012). Q-функція ($Q(s,a)$) оновлюється на кожному кроці моделювання, враховуючи нагороду та найкращі можливі дії в наступному стані, відповідно до рівняння

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max a' \, Q(s,a) - Q(s,a)],$$

де α – коефіцієнт навчання; γ – коефіцієнт дисконтування.

Застосування цього алгоритму в керуванні світлофорами дозволяє системі адаптивно регулювати сигнали на основі поточного стану трафіка, мінімізуючи затримки та підвищуючи ефективність дорожнього руху. Агент вибирає дії на основі стратегії, що балансує між дослідженням нових рішень і використанням відомих даних, причому нагорода обчислюється на основі зменшення часу затримок або черг на перехресті, що в кінцевому підсумку покращує загальну пропускну здатність транспортної системи. На кожному етапі моделювання (епоха), характеристики системи змінюють призначенням нагороди, якщо в результаті епохи час очікування сигналу світлофора менший за попередню епоху. І навпаки, нагороду не дають, якщо в результаті епохи час очікування сигналу світлофора більший за попередню епоху.

У цій роботі метод комп'ютерного зору використано для розпізнавання транспортних засобів. Застосовуючи модель YOLOv3 та бібліотеку OpenCV, система могла ідентифікувати типи та кількість транспортних засобів на перехрестях, що дозволило отримати необхідні дані для подальшої оптимізації роботи світлофорів.

Ці методи, інтегровані в межах єдиної системи, забезпечили ефективне керування транспортними потоками, дозволивши адаптувати сигнали світлофорів до реальних умов дорожнього руху, що значно зменшило затори і покращило загальну пропускну здатність міських доріг.

Математичну модель взаємодії компонентів системи представлено рівнянням оптимізації, де цільова функція мінімізує загальний час очікування на світлофорах для всіх учасників руху, а також враховує інтенсивність трафіка та затори.

Результати

На рис. 1 показано багаторівневу архітектуру IoT інтелектуальної транспортної системи.

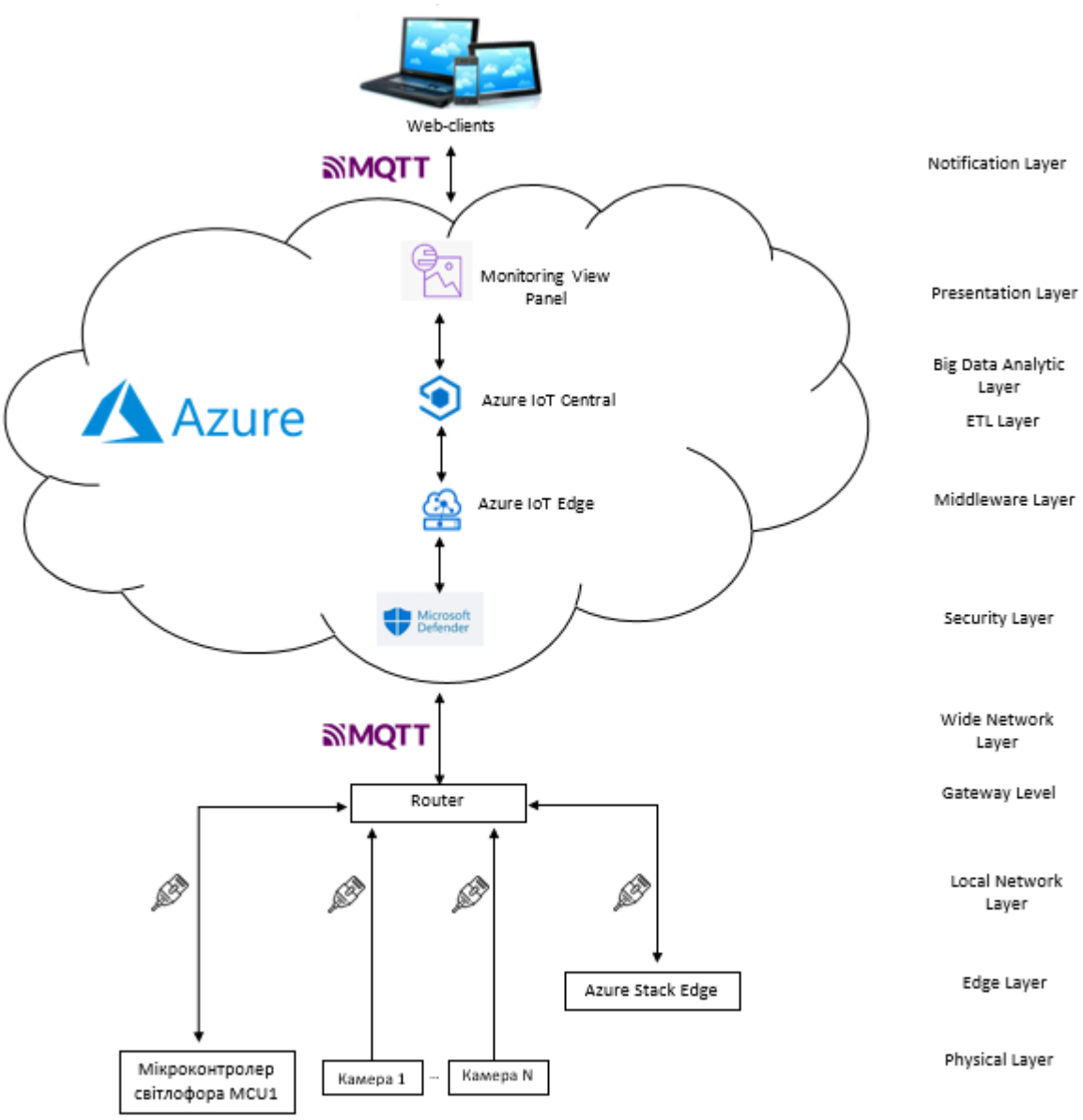


Рис. 1. Архітектура IoT інтелектуальної транспортної системи

1) Фізичний рівень. На цьому рівні розташовано камери, які в режимі реального часу записують відео про дорожню обстановку та передають дані до граничного пристрою Azure Stack Edge. Також на цьому рівні розташовано мікроконтролери світлофорів, які отримують команди від Azure Stack Edge і керують рухом на дорозі.

2) Рівень граничних обчислень. Тут розміщено пристрої Azure Stack Edge, які приймають відеозаписи з камер, розпізнають кількість транспортних засобів різних типів і передають дані в хмарне сховище, перетворюючи їх у формат JSON. Крім того, Azure Stack Edge запускає програму для моделювання дорожньої ситуації з метою оптимізації світлофорних сигналів.

3) Рівень периферійної комунікації. На цьому рівні дані, отримані від Azure Stack Edge, зводять до стандартного формату і передають до маршрутизатора.

ISSN 2788-6603

4) Рівень шлюзу. Azure Stack Edge виконує функції шлюзу для хмарного сховища, забезпечуючи автоматичну передачу даних до хмари та можливість локального доступу до файлів. Завдяки локальному кешу і регулюванню пропускної здатності для мінімізації ресурсоспоживання під час пікових навантажень, Azure Stack Edge ефективно оптимізує передачу даних між Azure і локальними пристроями.

5) Рівень зовнішньої комунікації. На цьому рівні дані від маршрутизатора передають у хмару Azure відповідно до стандартів і протоколів інтернету. Для обміну повідомленнями з тепловими пристроями використовують протокол MQTT, який працює за схемою "видавець-підписник". MQTT передає дані до сервісу Azure IoT Edge на хмарній платформі Azure, де зареєстровано всі пристрої транспортної системи.

6) Рівень безпеки. Microsoft Defender забезпечує цілодобовий моніторинг, керування та реагування на складні загрози, ризики й вимоги відповідності. Microsoft Azure пропонує різні рішення для зберігання даних, включаючи дискові, файлові сховища, а також сховища великих об'єктів і таблиць. Для захисту баз даних, таких як SQL Azure і Azure Cosmos DB, використовують шифрування й інші методи безпеки. Усі мережні з'єднання шифрують за допомогою TLS/SSL.

7) Рівень внутрішньої серверної комунікації. На цьому рівні відповідає PaaS сервіс Azure IoT Central, який забезпечує такі можливості:

- швидке створення додатків на основі сервісів;
- постійне оновлення додатків і забезпечення їхньої безперервної доступності;
- підтримка спеціалізованих моделей програмування і сервісів для конкретних завдань;
- висока керованість сервісів і додатків;
- оптимізація та еластичне масштабування відповідно до навантаження.

8) Рівень збору, оброблення та зберігання даних. Цей рівень обслуговує сервіс IoT Central, який збирає, обробляє та зберігає дані, отримані від Azure Stack Edge. На етапі збору дані копіюють або вилучають із вихідних джерел у проміжну область. Джерела можуть бути будь-якими – від SQL або NoSQL серверів до CRM і ERP систем, текстових файлів, електронних листів, вебсторінок тощо. У проміжній області дані трансформуються, щоб стати придатними для аналізу та відповідати схемі цільового сховища, яке зазвичай базується на OLAP або реляційних базах даних.

9) Рівень аналітики. Аналітику даних виконують за допомогою сценаріїв IoT Central, що дозволяє приймати, аналізувати дані і, в результаті, відправляти команди на мікроконтролери, а також відображати результати аналізу на панелі моніторингу.

10) Рівень сповіщень. Для інформування користувачів про дорожні ситуації використовують сценарії IoT Central, за допомогою яких можна надсилати повідомлення на мобільні пристрої через SMS або електронну пошту.

11) Рівень представлення. Представлення даних про систему виконують за допомогою IoT Central, який візуалізує інформацію у зручному для користувача вигляді, наприклад, у вигляді діаграм, графіків тощо.

Відповідно до вибраної архітектури, спроєктовано структурно логічну схему вузла інтелектуальної транспортної системи, яку зображено на рис. 2.

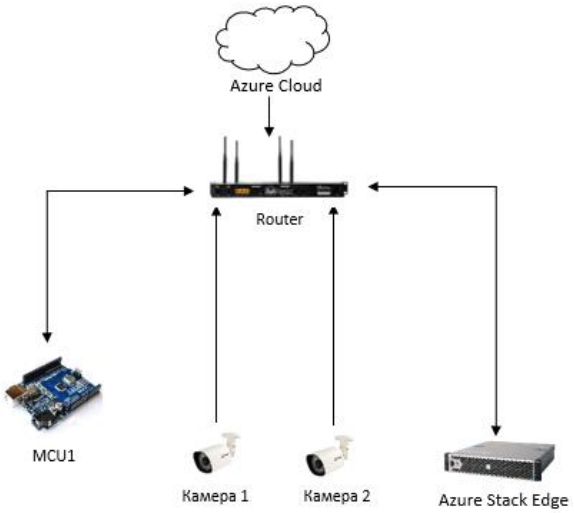


Рис. 2. Структурно логічна схема вузла інтелектуальної транспортної системи

У створеному вузлі системи розміщено мікропроцесор світлофора, пристрій Azure Stack Edge і камери відеоспостереження. Камери та пристрій Azure Stack Edge підключено до маршрутизатора через кабель Ethernet, а мікроконтролер взаємодіє з маршрутизатором по Wi-Fi за протоколом IEEE 802.11n. Максимальна відстань між компонентами вузла системи становить 15 м, що відповідає вимогам стандарту.

Процес отримання відео від камери та передачі даних про дорожню ситуацію у хмару через пристрій Azure Stack Edge включає такі етапи:

1) пристрій Azure Stack Edge через кожен заданий інтервал часу запитує відео про дорожню обстановку від камери;

2) використовуючи програму розпізнавання транспортних засобів, формуються дані про кількість різних типів транспорту на дорозі;

3) отримані дані формуються у JSON-повідомлення, яке містить ідентифікатор мікроконтролера, інформацію про наявність затору на дорозі, поточний колір світлофора, режим роботи світлофора та його локацію. Це повідомлення передається у хмару.

Отримано дані у вигляді JSON-повідомлення, яке складається з ідентифікатора мікроконтролера, виявлення затору на дорозі, колір світлофора на даний момент, режим світлофора та локацію відправляється у хмару.

На рис. 3 представлено блок-схему процесу отримання пристроєм Azure Stack Edge відео від камери та передачі даних про стан на дорозі у хмару.

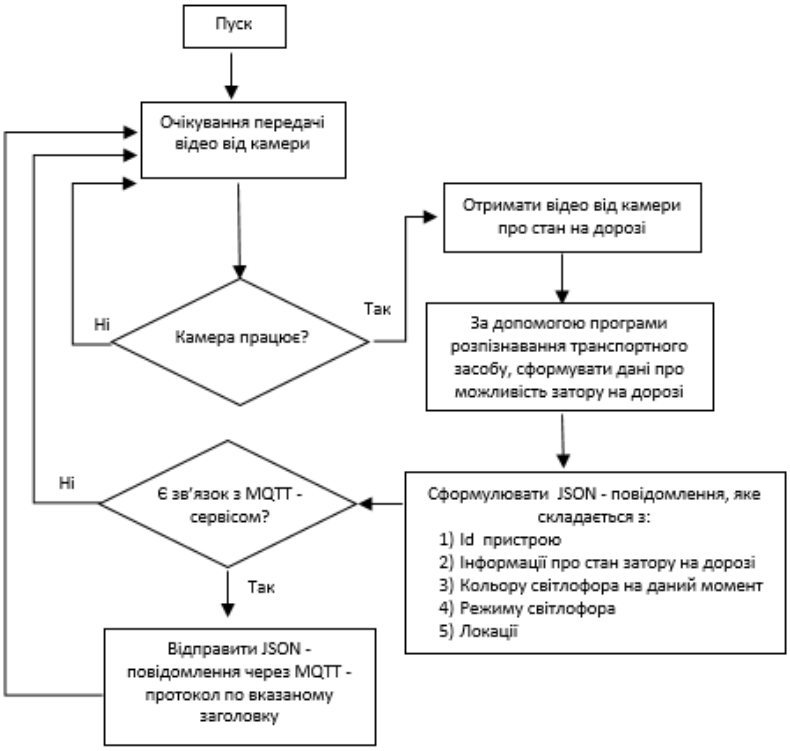


Рис. 3. Блок-схема процесу отримання пристроєм Azure Stack Edge відео від камери та передачі даних про стан на дорозі у хмару

Робота хмарного сервісу керування мережею IoT транспортної системи відбувається у такий спосіб:

1) Сервіс IoT Central отримує JSON-повідомлення від усіх пристроїв Azure Stack Edge, які складаються з відомостей щодо id пристрою, виявлення затору на дорозі, кольору світлофора на даний момент, режиму світлофора, локації. Отриману інформацію зберігають у хмарній БД Azure IoT Central.

2) Отриману інформацію надсилають до панелі моніторингу системи, де оператор може переглянути стан на дорозі.

3) В разі виникнення затору на дорозі оператор віддає команду пристрою Azure Stack Edge, що треба запустити програму оптимізації світлофорного регулювання.

Далі, коли необхідні параметри знайдено, пристрій Azure Stack Edge передає команду світлофору змінити час зеленого та червоного кольору.

На рис. 4 показано блок-схему хмарного сценарію, що відповідає за систему керування інтелектуальною транспортною мережею.

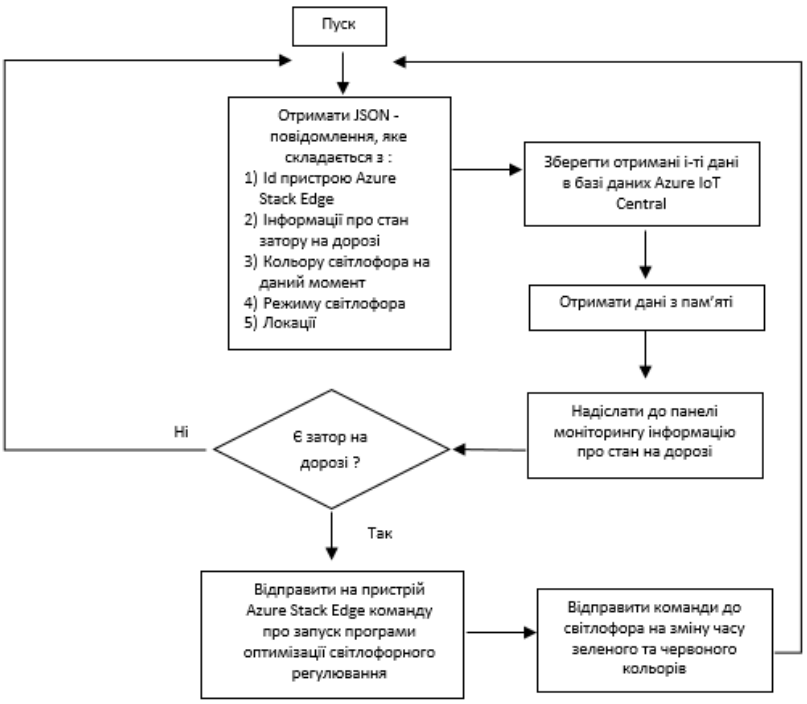


Рис. 4. Блок-схема хмарного сценарію, що відповідає за систему керування інтелектуальною транспортною мережею

ISSN 2788-6603

Для навчання моделі вибрано алгоритм навчання з підкріпленням (Q-Learning). На кожному етапі моделювання характеристики системи змінюють за допомогою призначення нагороди, якщо в результаті епохи час очікування сигналу світлофора менший за попередню епоху. І навпаки, нагороду не дають, якщо результат епохи часу очікування сигналу світлофора більший за попередню епоху.

На рис. 5 зображено блок-схему алгоритму навчання моделі транспортної системи.

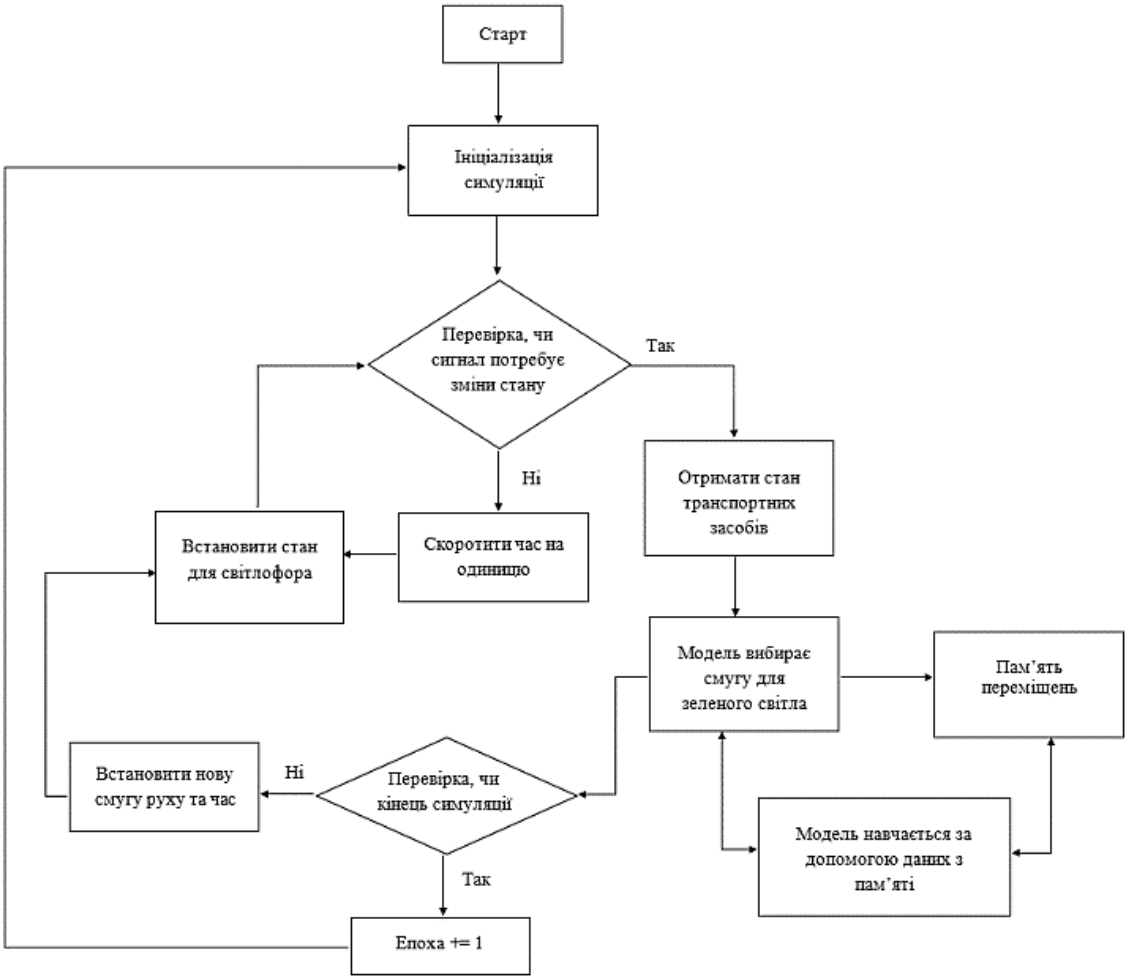


Рис. 5. Блок-схема алгоритму навчання моделі транспортної системи

Програма розпізнавання транспортних засобів використовує YOLOv3 у поєднанні з OpenCV для детекції та класифікації транспортних засобів у реальному часі. Процес включає:

1) Завантаження файлів моделі YOLOv3.
2) Налаштування моделі за допомогою OpenCV.
3) Оброблення відеокадрів для виявлення транспортних засобів.
4) Анотацію виявлених транспортних засобів із прямокутниками і класовими мітками.
5) Зберігання даних у форматах CSV та JSON для інтеграції з хмарою.

Результати. Програма успішно ідентифікує та класифікує транспортні засоби, надаючи дані про трафік у реальному часі (рис. 6).

Програма моделювання створює ситуації на дорозі й оптимізує час переключення світлофорів за допомогою алгоритму Q-Learning. Процес включає:

1) Створення фіксованих дорожніх подій для навчання моделі.
2) Налаштування часів світлофорів на основі кількості транспортних засобів і часу очікування.

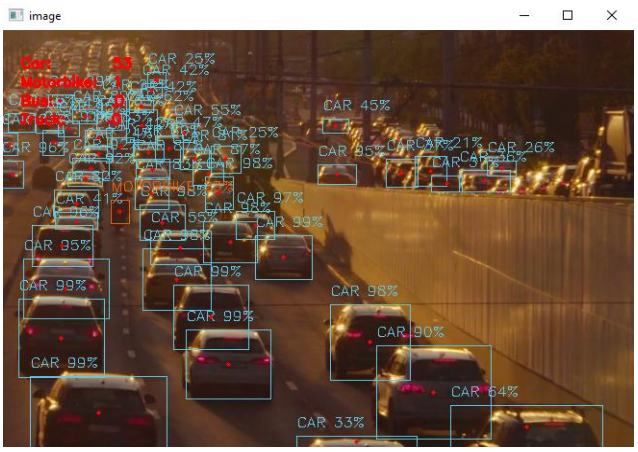


Рис. 6. Результат розпізнавання транспортних засобів

3) Використання бібліотек Optparse, Torch, Matplotlib і Traci для моделювання та візуалізації.
4) Виведення моделі оптимальних параметрів для кожної дорожньої ситуації.

Результати. Оптимізація привела до зменшення часу очікування до 20 % і зниження заторів на 15 % порівняно з традиційними системами.

На рис. 7 і 8 зображено приклади схем моделювання та результати навчання моделей.

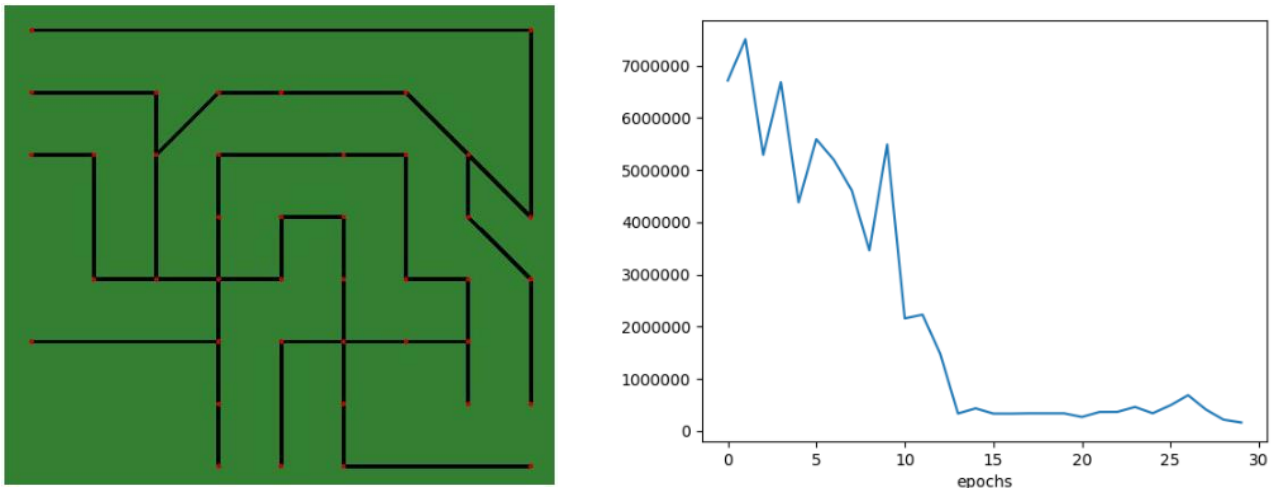


Рис. 7. Приклад 1 схеми моделювання та результати навчання моделі

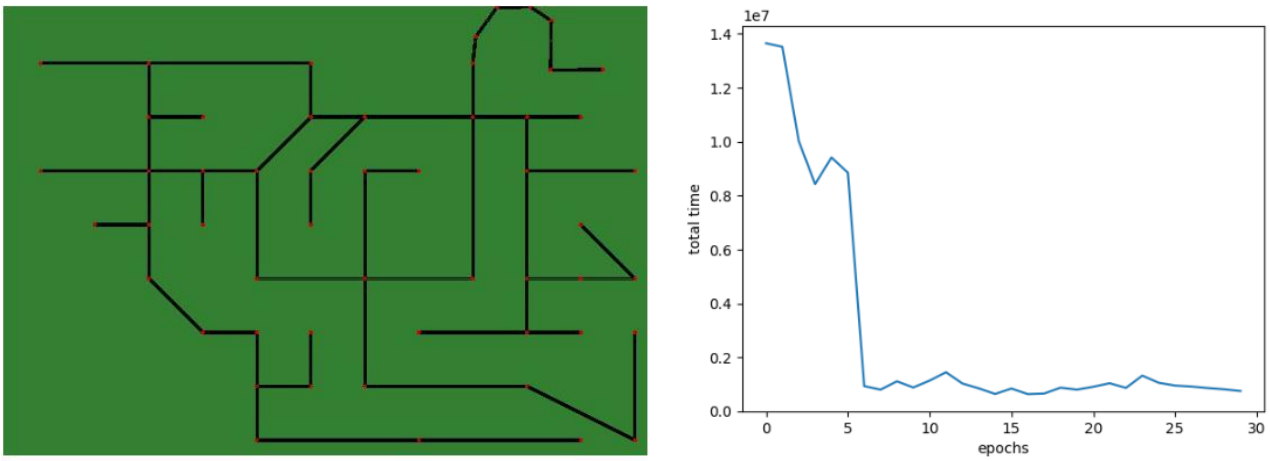


Рис. 8. Приклад 2 схеми моделювання та результати навчання моделі

Панель моніторингу розроблено з використанням Azure IoT Central для візуалізації та керування даними. Основні компоненти включають:

1) Шаблон пристрою: визначає характеристики та поведінку пристроїв.
2) Телеметрія: моніторинг стану світлофорів, їхнє місцезнаходження, події та кольори.
3) Команди: дозволяє реальний контроль режимів і часу сигналів світлофорів.
4) Властивості: відстежує ідентифікатори світлофорів.

Результати. Панель забезпечує реальний моніторинг стану світлофорів і дозволяє динамічно коригувати налаштування.

Систему тестували на стандартному комп'ютері замість Azure Stack Edge. Основні етапи включали:

1) Запуск програми моделювання світлофорів і підключення до Azure IoT Central.
2) Запуск програми розпізнавання транспортних засобів із прикладом відеокадру.
3) Спостереження за реальними умовами дорожнього руху на панелі моніторингу.
4) Експортування даних транспортної системи з OpenStreetMap і конвертація для SUMO.
5) Навчання моделі й аналіз результатів.

Результати. Система ефективно зменшила час очікування на світлофорах і оптимізувала рух транспорту за допомогою поєднання хмарного оброблення даних і IoT (рис. 9–13).

Інтеграція методів у систему здійснювалася через спільне використання інструментів комп'ютерного зору для розпізнавання транспортних засобів та алгоритмів машинного навчання для оптимізації керування світлофорами. Всі компоненти зв'язуються через хмарну платформу, що забезпечує їхню узгоджену роботу.

ISSN 2788-6603

Результати експериментів підтверджують, що запропонована архітектура забезпечує ефективне керування транспортними потоками в умовах житлового мікрорайону. Архітектура дозволяє оптимізувати роботу світлофорів, знижуючи затори і підвищуючи пропускну здатність на дорогах завдяки своєчасному обробленню даних і використанню алгоритмів машинного навчання.

Дискусія і висновки

Отже, основним науковим внеском роботи є розроблення та впровадження нової архітектури IoT системи для моніторингу та керування дорожнім трафіком. Запропонована система відрізняється високою адаптивністю, ефективністю в реальному часі та може бути застосована в різних умовах міського трафіка. Перспективи подальших досліджень полягають в удосконаленні алгоритмів навчання і їхній адаптації до різних транспортних ситуацій.

JamStatus

Timestamp	State name	Value
Now	Jam Status	Yes
Now	Jam Status	Yes
Now	Jam Status	Yes
Now	Jam Status	No
Now	Jam Status	No
Now	Jam Status	No

Рис. 9. Виникнення повідомлення про появу затору на перехресті

Рис. 10. Експортування частини транспортної системи

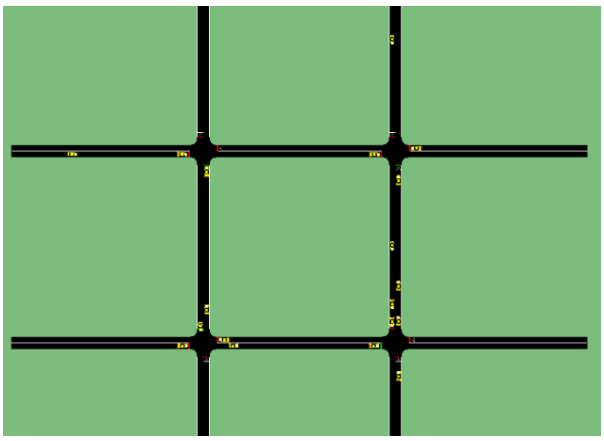


Рис. 11. Моделювання транспортної системи

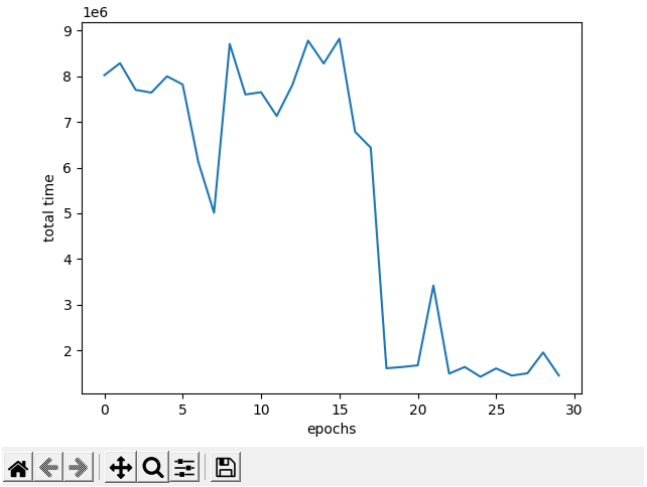


Рис. 12. Графік навчання моделі транспортної системи

Traffic Light / ChangeColorTime

Color&Time

red

Запустить

Рис. 13. Команда зміни часу сигналу світлофора

Внесок авторів: Олександр Сбітнєв – огляд літературних джерел, збір та аналіз емпіричних даних і проведення емпіричних досліджень, розроблення гібридної архітектури та проєктування системи, розроблення програмного забезпечення та його інтеграція з платформою Azure; Людмила Волощук – розроблення методології дослідження, використання хмарних сервісів Azure, написання висновків.

Список використаних джерел

Клюєв С.О., Цимбал С. В., & Сігонін А.Є. (2023). Розвиток інтелектуальних транспортних систем. *Вісник машинобудування та транспорту*, 2(18), 80–84. https://doi.org/10.31649/2413-4503-2023-18-2-80-86

Сбітнєв О. Ю. (2022). *Дослідження і розробка методів побудови хмарних систем керування IoT мережею* [Магістерська робота, Одеський національний університет імені І. І. Мечникова]. Одеса

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. *Journal of Intelligent Transportation Systems*, 18(3), 227–245.

Hunt, P. B., Robertson, D. I., & Bretherton, R. D. (1981). SCOOT - A traffic responsive method of coordinating signals. *TRL Laboratory Report*, 1014.

Katerna, O. Інтелектуальна транспортна система: проблема визначення та формування системи класифікації. *Економічний аналіз*, 29(2), 33–43.

Wiering, M., & van Otterlo, M. (2012). *Reinforcement learning: State-of-the-art.* Springer.

Gayratov, Z. K., Kilichov, J. R., & Toshpulatov, A. (2022). Basic definitions of twelve layer IoT architecture for smart city. *In International Scientific and Technical Conference: Digital Technologies: Problems and Solutions of Practical Implementation in the Industry*. Belarusian State University of Informatics and Radioelectronics,

Danshyna, S. Yu., Nechausov, A. S., & Andrieiev, S. M. (2022). Information technology of transport infrastructure monitoring based on remote sensing data. *Радіоелектроніка, інформатика, управління*, 4, 7–14. https://doi.org/10.15588/1607-3274-2022-4-7

References

El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2014). Design of reinforcement learning parameters for seamless application of adaptive traffic signal control. Journal of Intelligent Transportation Systems, 18(3), 227-245.

Hunt, P. B., Robertson, D. I., & Bretherton, R. D. (1981). SCOOT - A traffic responsive method of coordinating signals. TRL Laboratory Report, 1014.

Katerna, O. (2019). Intelligent transport system: The problem of definition and the formation of the classification system. Economic Analysis, 29(2), 33-43.

Klyuev, S. O., Tsymbal, S. V., & Sihonin, A. E. (2023). Development of intelligent transport systems. Bulletin of Mechanical Engineering and Transport, 2(18), 80-84. https://doi.org/10.31649/2413-4503-2023-18-2-80-86

Sbitnev, O. Yu. (2022). Research and development of cloud-based control systems for IoT networks (Master's thesis). Odessa.

Wiering, M., & van Otterlo, M. (2012). Reinforcement learning: State-of-the-art. Springer.

Gayratov, Z. K., Kilichov, J. R., & Toshpulatov, A. (2022). Basic definitions of twelve layer IoT architecture for smart city. In International Scientific and Technical Conference: Digital Technologies: Problems and Solutions of Practical Implementation in the Industry.

Danshyna, S. Yu., Nechausov, A. S., & Andrieiev, S. M. (2022). Information technology of transport infrastructure monitoring based on remote sensing data. *Радіоелектроніка, інформатика, управління*, 4, 7–14. https://doi.org/10.15588/1607-3274-2022-4-7

Отримано редакцією журналу / Received: 21.08.24
Прорецензовано / Revised: 19.09.24
Схвалено до друку / Accepted: 07.11.24

Lyudmila VOLOSHCHUK, PhD (Engin.), Assoc. Prof.
ORCID ID: 0000-0002-2510-0038
e-mail: lavstumbre@gmail.com
Odessa I. I. Mechnikov National University, Odesa, Ukraine

Oleksandr SBITNEV, PhD Student
ORCID ID: 0009-0008-6311-612X
e-mail: alexsbitnev99@gmail.com
Odessa I. I. Mechnikov National University, Odesa, Ukraine

HYBRID CLOUD-BASED INTELLIGENT TRAFFIC MONITORING IOT SYSTEM FOR A RESIDENTIAL AREA

Background. ***This paper presents a new architecture for an intelligent transportation system (ITS) that leverages Internet of Things (IoT) technologies and the Azure cloud platform. The scientific novelty lies in the development of an architecture that integrates edge computing, cloud services, and machine learning algorithms for adaptive traffic management based on real-time data. The proposed architecture efficiently processes traffic flow information, performs modeling, and automatically adjusts traffic signals to reduce congestion. The effectiveness of the architecture has been validated through a series of experiments focused on vehicle recognition, traffic signal optimization, and real-time monitoring of the traffic situation.***

Methods. ***The methods used include computer simulation modeling for managing the intelligent transportation system, reinforcement learning for training the system, and computer vision techniques for vehicle recognition.***

Results. ***The proposed ITS architecture is based on IoT technologies, enabling real-time data collection and analysis of road traffic. The developed system was tested in various urban areas with different levels of traffic load. The experiments demonstrated that the system can adaptively adjust traffic signals based on traffic analysis, significantly improving road capacity and reducing congestion.***

Conclusions. ***The results of the experiments confirmed the effectiveness of the proposed intelligent transportation system architecture. Future research may focus on enhancing the system by incorporating more advanced artificial intelligence algorithms for automating traffic signal management decisions.***

Keywords: ***intelligent transportation system, cloud platform, transportation system optimization, IoT system architecture design.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.

ISSN 2788-6603

ІНФОРМАЦІЯ ПРО АВТОРІВ / INFORMATION ABOUT AUTHORS

Наталія Аксак, д-р техн. наук, проф., професор кафедри комп'ютерних інтелектуальних технологій та систем факультету комп'ютерної інженерії та управління Харківського національного університету радіоелектроніки.
Наукові інтереси: інтелектуальні комп'ютерні системи, паралельні та розподілені обчислення, мультиагентні системи.

Natalia Aksak, DSc (Engin.) Prof., Professor of the Department of Computer Intelligent Technologies and Systems, Faculty of Computer Engineering and Control, Kharkiv National University of Radio Electronics.
Research interests: intelligent computer systems, parallel and distributed computing, multi-agent systems.

Олег Барабаш, д-р техн. наук, проф., професор кафедри інженерії програмного забезпечення в енергетиці Навчально-наукового інституту атомної і теплової енергетики Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: функціональна стійкість інформаційних систем, технічне діагностування.

Oleg Barabash, DSc (Engin.), Prof., Professor of Department of Software Engineering for Power Industry of Institute of Nuclear and Thermal Energy, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" of Kyiv.
Research interests: functional stability of informational systems, technical diagnostics.

Майя Бочарова, аспірантка, кафедра математичного забезпечення комп'ютерних систем, Одеський національний університет імені І. І. Мечникова.
Наукові інтереси: природне оброблення мови; пошук подібності; навчання машин; аналіз даних, інженерія програмного забезпечення.

Maiia Bocharova, PhD Student, Department of Mathematical Support of Computer Systems, Odesa I. Mechnikov National University
Research interests: Natural Language Processing, similarity search, machine learning, data mining, software engineering.

Олексій Бичков, д-р техн. наук, професор кафедри програмних систем та технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.
Наукові інтереси: наука про дані, кібернетично-фізичні системи, математичне моделювання, нечітке моделювання, розроблення програмного забезпечення, розроблення і впровадження інформаційних систем.

Oleksii Bychkov, DSc (Engin.), Prof., Professor of the Department of Software Systems and Technologies, Faculty of Information Technologies, Taras Shevchenko National University of Kyiv.
Research interests: data science, cyber-physical systems, mathematical modeling, fuzzy modeling, software development, information systems development and implementation.

Денис Бородай, аспірант факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.
Наукові інтереси: програмування на Python, штучний інтелект, аналіз даних.

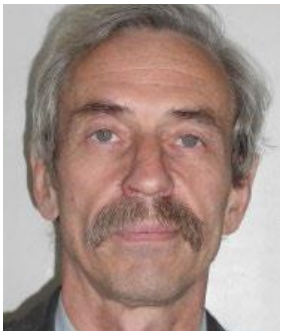
Denys Borodai, PhD Student, Faculty of Information Technologies, Taras Shevchenko National University of Kyiv.
Research interests: computer programming on Python, artificial intelligence, data analysis.

Людмила Волощук, канд. техн. наук, доцент кафедри математичного забезпечення комп'ютерних систем Одеського національного університету імені І. І. Мечникова.
Наукові інтереси: комп'ютерні мережі, хмарні технології, технології захисту інформації, машинне навчання, big data.

Lyudmila Voloshchuk, PhD (Engin.), Associate Professor of the Department of Mathematical Support of Computer Systems, Odessa I. I. Mechnikov National University.
Research interests: computer networking, cloud technology, information security technology, machine learning, big data.

Олександра Дмитренко, аспірантка, 4-го року навчання інформаційних технологій в телекомунікаціях Навчально-наукового інституту телекомунікаційних систем, асистент Навчально-наукового інституту атомної і теплової енергетики Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: програмування на Java та Python, алгоритми та структури даних, бази даних.

Oleksandra Dmytrenko, PhD Student of the Information Technologies in Telecommunications Department, Educational and Scientific Institute of Telecommunication Systems, assistant of Educational and Scientific Institute of Atomic and Thermal Energy at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute."
Research interests: computer programming in Java and Python, algorithms and data structures, databases.



Євген Івохін, д-р фіз.-мат. наук, проф., професор кафедри системного аналізу та теорії прийняття рішень факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка.
Наукові інтереси: нечіткі моделі та системи, моделі розповсюдження інформації, оптимізаційні задачі.

Eugene Ivohin, DSc (Phys. & Math.), Prof., Professor of the Department of System Analysis and Decision Making, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv.
Research interests: fuzzy models and systems, information distribution models, optimization problems.

Олексій Кузнєцов, аспірант кафедри системного проєктування Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: штучний інтелект, проєктний менеджмент, бізнес-аналітика.

Oleksii Kuznietsov, PhD Student of the Department of System Design at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
Research interests: artificial intelligence, project management, business analyst.

Юрій Кравченко, д-р техн. наук, проф., завідувач кафедри мережевих та інтернет технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.
Наукові інтереси: телекомунікаційні системи, теорія штучного інтелекту.

Yurii Kravchenko, DSc (Engin.), Prof., Head of the Department of Networking and Internet Technologies, Faculty of Information Technologies, Taras Shevchenko National University of Kyiv.
Research interests: telecommunication systems, theory of artificial intelligence.

ISSN 2788-6603

Геннадій Кисельов, канд. техн. наук, доц., старший науковий співробітник і заступник завідувача кафедри системного проєктування Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: математичне моделювання цифрових схем і систем, створення систем автоматизованого проєктування, створення програмно-методичного забезпечення систем дистанційного навчання та мультимедіа застосувань, програмування розподілених систем і Grid застосувань.

Gennadiy Kyselov, PhD (Engin.), Assoc. Prof., Senior Researcher and Deputy Head of the Department of System Design at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".
Research interests: mathematical modeling of digital circuits and systems, development of computer-aided design systems, creation of software and methodological support for distance learning systems and multimedia applications, programming of distributed systems and Grid applications.

Максим Кушнарьов, канд. техн. наук, старший викладач кафедри комп'ютерних інтелектуальних технологій та систем факультету комп'ютерної інженерії та управління Харківського національного університету радіоелектроніки.
Наукові інтереси: інтелектуальні системи оброблення музики, адаптивні системи керування БпЛА.

Maksym Kushnaryov, PhD (Engin.), Senior Lecturer of the Department of Computer Intelligent Technologies and Systems, Faculty of Computer Engineering and Control, Kharkiv National University of Radio Electronics.
Research interests: intelligent music processing systems, adaptive UAV control systems.

Андрій Макарчук, аспірант кафедри інженерії програмного забезпечення в енергетиці Навчально-наукового інституту атомної і теплової енергетики Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: функціональна стійкість, машинне навчання з учителем, методи наближення, математичне моделювання.

Andrii Makarchuk, PhD student of the Department of Software Engineering For Power Industry of Institute of Nuclear and Thermal Energy, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" of Kyiv.
Research interests: functional stability, supervised machine learning, approximation methods, mathematical modeling.

Євгеній Малахов, д-р техн. наук, проф., завідувач кафедри математичного забезпечення комп'ютерних систем, Одеський національний університет імені І. І. Мечникова.
Наукові інтереси: теорія баз даних, метамоделювання, методи аналізу даних та інших структур даних, методи оброблення даних.

Eugene Malakhov, DSc (Engin.), Prof., Head of Department of Mathematical Support of Computer Systems, Odesa I. Mechnikov National University.
Research interests: databases theory; metamodeling, the methods of data mining and other data structuring, data processing methods.

Микола Мороз, аспірант кафедри програмних систем та технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.
Наукові інтереси: вбудовані системи, операційні системи, C++, програмні оптимізації коду.

Mykola Moroz, PhD Student of the Department of Software Systems and Technologies, Faculty of Information Technologies, Taras Shevchenko National University of Kyiv.
Research interests: embedded systems, operation systems, C++, software code optimizations.

Віталій Омельченко, аспірант кафедри інформаційних систем та технологій факультету інформатики та обчислювальної техніки Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського"
Наукові інтереси: розроблення розподілених систем і інструментів для них, машинне навчання.

Vitalii Omelchenko, PhD Student of the Department of Informational System and Technologies, Faculty of Informatics and Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".
Research interests: development of distributed systems and tools, machine learning.

Олександр Ролік, д-р техн. наук, проф., професор кафедри інформаційних систем та технологій факультету інформатики та обчислювальної техніки Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: управління IT-інфраструктурами, методи та засоби надання інформаційних сервісів.

Oleksandr Rolik, DSc (Engin.), Prof., Professor of the Department of Informational System and Technologies, Faculty of Informatics and Computer Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".
Research interests: management of IT-infrastructure, methods and tools for providing information services.

Олександр Сбітнєв, аспірант кафедри математичного забезпечення комп'ютерних систем Одеського національного університету імені І. І. Мечникова.
Наукові інтереси: комп'ютерні мережі, хмарні технології, туманні технології, машинне навчання, інтернет речей, аналіз даних.

Oleksandr Sbitnev, PhD Student of the Department of Mathematical Support of Computer Systems, Odesa I. Mechnikov National University.
Research interests: computer networks, cloud technologies, nebulous technologies, machine learning, internet of things, data analysis.

Марія Скулиш, д-р техн. наук, завідувачка кафедри інформаційних технологій в телекомунікаціях Навчально-наукового інституту телекомунікаційних систем Національного технічного університету України "Київський політехнічний інститут імені Ігоря Сікорського".
Наукові інтереси: телекомунікаційні системи, мережі мобільного зв'язку, хмарні технології.

Mariia Skulysh, DSc (Engin.), Head of Information Technologies in Telecommunications Department, Educational and Scientific Institute of Telecommunication Systems at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute."
Research interests: telecommunication systems, mobile networks, cloud systems.

Юрій Шеліхов, аспірант кафедри комп'ютерних інтелектуальних технологій та систем факультету комп'ютерної інженерії та управління Харківського національного університету радіоелектроніки.
Наукові інтереси: інтелектуальна сіті-ферма.

Yurii Shelikhov, PhD Student of the Department of Computer Intelligent Technologies and Systems, Faculty of Computer Engineering and Control, Kharkiv National University of Radio Electronics.
Research interests: intellectual city-farm.

ISSN 2788-6603

Костянтин Юштін, канд. фіз.-мат. наук, докторант кафедри системного аналізу та теорії прийняття рішень факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка.
Наукові інтереси: оптимізаційні задачі з нечіткими параметрами, обчислювальні методи, програмування.

Kostyantyn Yushtin, PhD (Phys. & Math.), Doctoral Student of the Department of System Analysis and Decision Making, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv.
Research interests: optimization problems with fuzzy parameters, computational methods, programming.

Наукове видання

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

№ 1(3)/2024

Редактор *Л. Магда*

Оригінал-макет виготовлено Видавничо-поліграфічним центром "Київський університет"

Формат 60x84$^{1/16}$. Обл.-вид. арк. 10.3. Ум. друк. арк. 11,8. Наклад 100. Зам. № 224-11222.
Гарнітура Arial. Папір офсетний. Друк офсетний. Вид. № ІТ3.
Підписано до друку 28.02.25

Видавець і виготовлювач
ВПЦ "Київський університет",
Б-р Тараса Шевченка, 14, м. Київ, 01601, Україна
☎ (38044) 239 32 22; (38044) 239 31 58; (38044) 239 31 28
e-mail: vpc@knu.ua; vpc_div.chief@univ.net.ua; redaktor@univ.net.ua
http: vpc.univ.kiev.ua
Свідоцтво суб'єкта видавничої справи ДК № 1103 від 31.10.02