

Міністерство освіти і науки України

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНИЙ
УНІВЕРСИТЕТ

**МАТЕРІАЛИ
ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ
КОНФЕРЕНЦІЇ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ
ТА МОЛОДИХ УЧЕНИХ**

**«СУЧАСНИЙ СТАН ДОСЛІДЖЕНЬ
В СФЕРІ ІТ»**

(19 квітня 2024 року)

Харків 2024

ЗМІСТ

Секція 1. WEB-ТЕХНОЛОГІЇ, ХМАРНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ У ТРАНСПОРТНИХ ТА ПРОМИСЛОВИХ ЗАСТОСУВАННЯХ

<i>Babenko V.O., Kanishov V.I.</i> Architecture and approaches of cloud computing in transport systems: problems and opportunities	6
<i>Babenko V.O., Kovtun Ye.S.</i> Architecture of cloud service models: problems and perspectives	8
<i>Petruk Viktoriia</i> What can you create with the instagram api. Restrictions vs capabilities. Ai adoption	11
<i>Гриценко К.М.,</i> Вдосконалення створення форм на базі бібліотеці Reactjs.....	17
<i>Карпішен Б.С.</i> Використання технологій C-V2X ТА DSRC на автономному транспорті.....	21
<i>Сірий Д.О.</i> Розробка телеграм бота для пасажирських перевезень.....	27
<i>Кудінов Є.О.,</i> Реалізація внутрішньої маршрутизації між роутерами з кількома провайдерами на основі Routers.....	30
<i>Бабенко В.О., Лантрат О.В.</i> Розвиток інформаційно-комунікаційної платформи для процесів управління документацією транспортних мереж	35
<i>Масін І.Е.</i> Моделювання процесу обслуговування клієнтів у банку	37
<i>Олена Шапошнікова О., Олійник Д.</i> Інформаційна система вибору станції технічного обслуговування.....	41
<i>Полупан Анна Віталіївна, Пронін С.В.</i> Розробка Web-додатку по проходженню тестів правил дорожнього руху	43
<i>Соколова А.О., Пронін С.В.</i> Покращення системи державного керування за рахунок застосування технології блокчейнш.....	45
<i>Шапошнікова О., Степанов Г.</i> Використання технології REACT.JS для створення сучасних Веб застосунків	49
<i>Стефанців А.В., Лебединський А.В.</i> Аналіз найпопулярніших CMS для створення інтернет-магазинів	52

Хоменко Ю.С., Інформаційна технологія автономної навігації на місцевості.....	56
Шешунов М.О., Пронін С.В. Удосконалення сервісів у сфері оренди та купівлі житла за допомогою інформаційних технологій.....	58
Бабенко В.О., Тимченко С.С. Проблеми та перспективи систем пошуку з використанням хмарних технологій.....	61
Мацій О.Б., Фентисов Б. Огляд основних етапів проектування сайту.....	63
Портніхін К.О. Телеграм бот для магазину одягу.....	65
Балабай А.О., Керівник Плехова Г.А. Кібербезпека у хмарних технологіях.....	69

Сесія 2. ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ УПРАВЛІННЯ

Вабенко V.O., Drozdyk Ye.V. Applying of big data analytics in intelligent transport systems.....	72
Барашков В.С., Пронін С.В. Інформаційна система для побудови логістичних ланцюгів.....	75
Грицук В.Ю., Пронін С.В. Перспективи впровадження прогностичного обслуговування транспортних засобів на основі машинного навчання.....	77
Петренко Ю.А., Жабін О.Ю., Аналіз стандартної автоматизації бізнес процесів підприємств та автоматизації з використанням штучного інтелекту.....	82
Зубко М.В., Лебединський А.В. Використання «ТЕХТ-ТО-SPEECH» моделі AUDIOAPI від OPENAI.....	86
Кирилов Д.І., Лебединський А.В. Використання генеративної моделі штучного інтелекту DALL-E у вебдодатках.....	88
Кір'яков В.Д. Використання нейронних мереж у медицині.....	91
Літвін О., Пронін С.В. Глибокі мовні моделі: від технологічного прориву до практичних застосувань.....	95
Мізяк І. О. Модель інтелектуальної інформаційної системи управління транспортною системою міста.....	99
Решитько В.С., Карпішен Б.С. Використання нейронних мереж у розробці дизайну сайтів.....	102
Соколовський А.О. Нейронна мережа для прогнозування часових рядів.....	107

<i>Товстолуг Д. В.</i>	
Нейронна мережа для прогнозування тональності тексту.....	111
<i>Халимон В.К.</i>	
Дослідження та аналіз використання нейронної мережі для розпізнавання рукописних цифр.....	115
<i>Харченко С.Д.</i>	
Використання нейронної мережі для синтемент-аналізу	120
<i>Черняк О.К.</i>	
Розробка нейронної мережі для сегментації об'єктів дорожнього руху ...	125
<i>Шпіль Є., Пронін С.В.</i>	
Використання ШІ у сучасному дизайні	127

Секція 3. КОМП'ЮТЕРНА, ПРОГРАМНА, СИСТЕМНА ІНЖЕНЕРІЯ ІТ-ІНДУСТРІЇ

<i>Гриценко К.М., Петренко Ю.А.,</i>	
Розробка методичного комп'ютерного комплексу підготовки операторів диспетчерського управління асфальто бетонними установками на основі технології НМІ.....	136
<i>Батура М.В.</i>	
Телеграм бот помічник подорожувальника	140
<i>Шапошнікова О., Білик А.</i>	
Методи знаходження ключових словосполучень.....	145
<i>Клячко М.М., Лебединський А.В.</i>	
Створення додатків на Windows Forms: архітектура та функціональні можливості	148
<i>Лаптєв Д.</i>	
Рекомендаційні системи та їх моделі	152
<i>Шапошнікова О., Сещенко І.</i>	
Розроблення іс соціальної взаємодії. Модуль «Соціальна адаптація»	153
<i>Шапошнікова О., Степанов Г.</i>	
Демістифікація візуалізації часових рядів: використання технології REACT.JS у проекті Storage.Timeline.....	156
<i>Шамельов В.О., Пронін С.В.</i>	
Розробка бота-перекладача	158
<i>Щербініна Ю. В., Лебединський А. В.</i>	
Розробка Android-додатків на Java та Kotlin: переваги та недоліки	162

Секція 4. ПРОБЛЕМИ ІНТЕГРАЦІЇ НАУКОВИХ ДОСЛІДЖЕНЬ, ОСВІТИ, ВИРОБНИЦТВА

<i>Бегунов Олександр</i>	
Сучасні ризики у транспорті: виклики та шляхи подолання.....	166

<i>Петренко Ю.А., Бугаєвський М.С.</i> Використання гнучких методів у проєктній діяльності з розробки програмного забезпечення.....	168
<i>Комков М.А.</i> Розробка телеграм бота для управління пасажирськими перевезеннями	173
<i>Шапошнікова О., Алексеєв Е.</i> Розроблення концепції системи електронного навчання.....	177

Секція 5. ФУНДАМЕНТАЛЬНІ ТА ПРИКЛАДНІ ДОСЛІДЖЕННЯ У ГАЛУЗІ ІТ-ТЕХНОЛОГІЙ І МЕХАТРОНІКИ

<i>Шапошнікова О., Земляной О.</i> Переваги впровадження інформаційних систем в управління автопарком	180
<i>Мацій О.Б., Клочко А.В.</i> Телеграм як інструмент для детального аналізу користувацької взаємодії	183
<i>Шапошнікова О., Насатов О.</i> Переваги впровадження «Eco friendly» у автоматизацію системи замовлення	185
<i>Онищенко Я.</i> Системний аналіз системи клієнт-банк.....	187
<i>Мацій О.Б., Романов М.В.</i> Аналіз ефективності процедурної генерації лабіринтів для мультиплеєрних 3D Веб-ігор на базі THREE.js	189
<i>Соболев Д.</i> Огляд систем масового обслуговування змішувач періодичної дії.....	191
<i>Шапошнікова О., Халимон В.</i> Аналітичний огляд систем автоматизації планування кошторису	192
<i>Хомутенко Д.Г.</i> Комп'ютерне моделювання структури асфальтобетону для дослідження його властивостей під дією низьких температур	195
<i>Чмутов А.</i> Підхід до системного аналізу імітаційної моделі глобального потепління	201
<i>Чорнобай Е.І., Карпішен Б.С.</i> Як штучний інтелект змінить автотранспортні вантажоперевезення	202
<i>Кривінченко Д. Р., Мельников О. Ю.</i> Прогнозування ефективності просування сайту за допомогою мови програмування та аналізу даних R	207

UDC 004.773

**ARCHITECTURE AND APPROACHES OF CLOUD
COMPUTING IN TRANSPORT SYSTEMS:
PROBLEMS AND OPPORTUNITIES**

Babenko V.O., Kanishov V.I.

Kharkiv National Automobile and Highway University, Kharkiv

Recently it is discussed concepts, developments, and categorizes the most relevant research and development efforts in cloud computing, particularly focusing on public clouds, management tools, and development frameworks. The introduction also highlights practical realizations of cloud computing, emphasizing architectural aspects and innovative technical features.

This model offers advantages to both IT service consumers and providers. Consumers can experience cost savings by opting for more affordable services from external providers instead of making significant investments in IT infrastructure and personnel.

The "on-demand" nature of this model enables consumers to adjust their IT usage according to rapidly changing or unpredictable computing requirements. On the other hand, IT service providers benefit from improved operational costs. By building hardware and software infrastructures that cater to multiple solutions and serve numerous users, providers can enhance efficiency, leading to faster return on investment (ROI) and reduced total cost of ownership (TCO).

Cloud computing presents a distinct value proposition compared to traditional enterprise IT environments. By leveraging virtualization and aggregating computing resources, it offers economies of scale that would otherwise be inaccessible. It also provides immediate opportunities to utilize existing hardware and software, eliminating the need for designing, deploying, and testing new implementations, which saves time and resources.

One of the key advantages of cloud computing is that it allows organizations to treat infrastructure and its management as operational expenses rather than capital investments. This has tax benefits and enables businesses to conserve capital for other purposes. Additionally, cloud computing provides a centralized and remote computing facility, leading to economies of scale in hardware and software usage and reducing the administrative resources required for management.

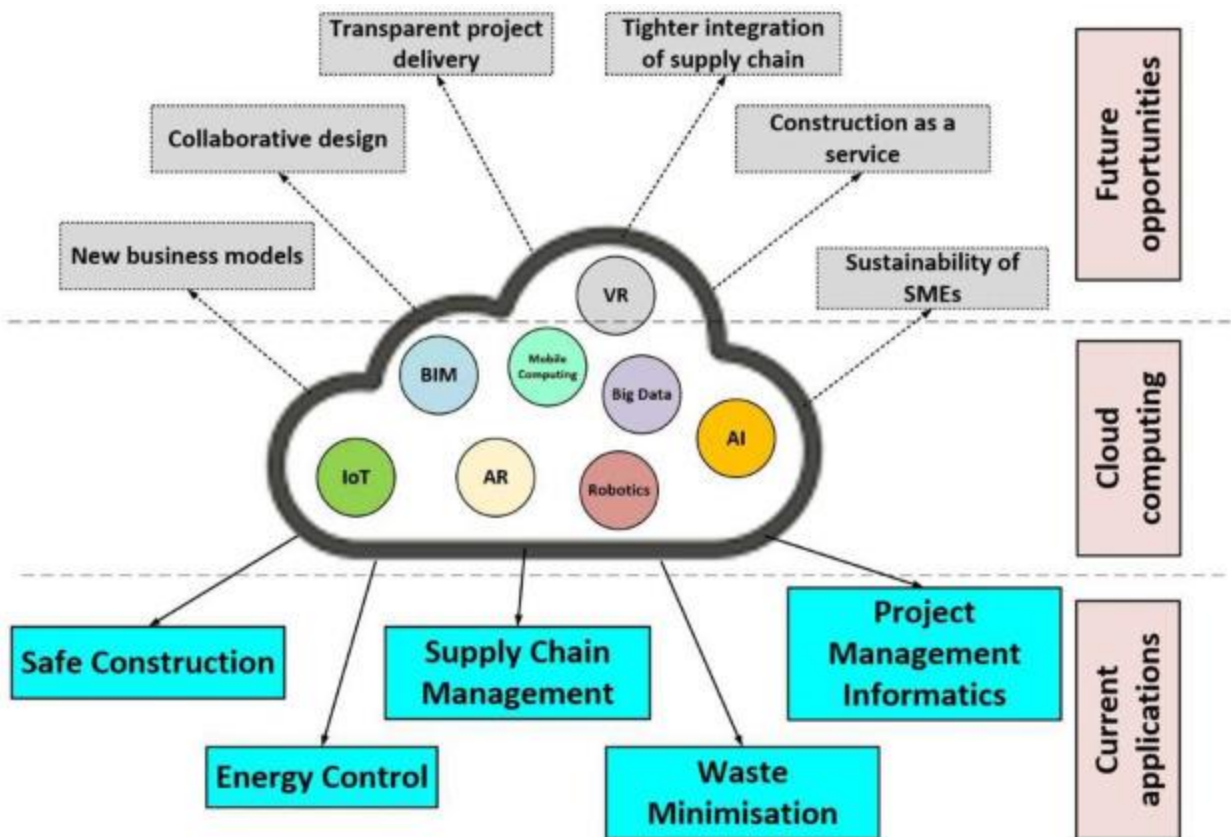


Figure 1. Depicts the convergence of multiple advancements that have ultimately led to the emergence of cloud computing

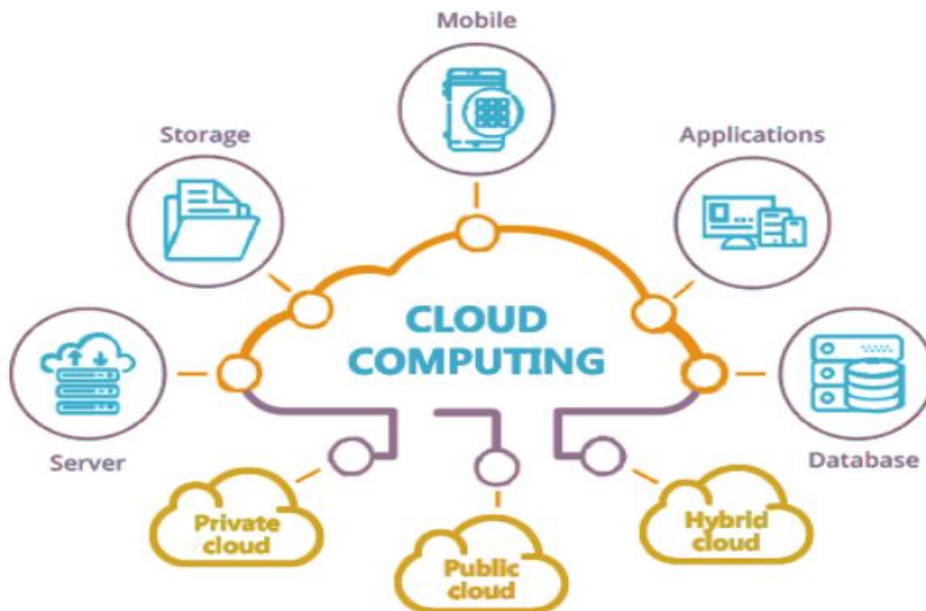


Figure 2. Cloud computing

The ability to access computing resources instantly, without the need for extensive time and skilled resources to set up infrastructure, results in faster time to

value. This can lead to enhanced revenue, increased business agility, a larger market share, and other benefits.

It's important to note that cloud computing doesn't exist in isolation. Most organizations already have a diverse range of applications running in their data centers. Cloud computing typically extends existing infrastructure, either by using it primarily for new projects or as overflow capacity to ensure a certain level of performance for enterprise computing.

The architecture and approaches of cloud computing involve the systems architecture of software systems used in delivering cloud services. Typically, this architecture includes multiple cloud components that communicate with each other using loose coupling mechanisms, such as messaging queues. The concept of elastic provision involves intelligent usage of tight or loose coupling mechanisms and other techniques.

Reference:

1. N. Carr, *The Big Switch: Rewiring the World, from Edison to Google*. W. W. Norton & Co., New York, 2020.

2. The Transformation of Education through State Education Clouds, www.ibm.com/ibm/files/N734393J24929X18/EBW03002-USEN-00.pdf

3. The Research and Application of Network Teaching Platform Based on Cloud Computing, Zhang Tao and Jiao Long, *International Journal of Information and Education Technology*, Vol. 1, No. 3, August 2011

4. Cloud Computing For Distributed University Campus: A Prototype Suggestion, http://www.pixelonline.net/edu_future/common/download/Paper_pdf/ENT30-Erkoc.pdf

UDC 004.773

ARCHITECTURE OF CLOUD SERVICE MODELS: PROBLEMS AND PERSPECTIVES

Babenko V.O., Kovtun Ye.S.

Kharkiv National Automobile and Highway University, Kharkiv

Typically, the architecture includes multiple cloud components that communicate with each other using loose coupling mechanisms, such as messaging queues. The architecture and approaches of cloud computing involve the systems architecture of software systems used in delivering cloud services. The concept of elastic provision involves intelligent usage of tight or loose coupling mechanisms and other techniques.

The main architecture of cloud computing can be categorized as follows:

A. Essential Characteristics of Cloud Computing:

- On-demand self-service: Users can provision computing resources without the need for human intervention.

- Broad network access: Services are accessible over a network through various devices.
- Resource pooling: Computing resources are pooled together and shared among multiple users, leading to efficiency and optimization.
- Location independence: Users can access cloud services regardless of their geographical location.
- Measured service: Cloud usage is monitored, controlled, and reported, allowing for transparent and metered billing.

B. Cloud Service Models:

- Software as a Service (SaaS): Users can utilize applications provided by the cloud service provider over a network.
- Platform as a Service (PaaS): Users can deploy their own applications onto the cloud infrastructure provided by the service provider.
- Infrastructure as a Service (IaaS): Users can rent processing power, storage, and network capacity from the service provider.

C. Cloud Deployment Models:

- Public Cloud: Infrastructure is owned and operated by a cloud service provider and is available to the general public.
- Private Cloud: Infrastructure is owned or leased by a single enterprise and is dedicated to its specific needs.
- Hybrid Cloud: Combination of public and private clouds, allowing for flexibility and customization.
- Community Cloud: Shared infrastructure designed for a specific community or group with common requirements.

These architecture and deployment models provide different options for organizations to choose from based on their specific needs, preferences, and security requirements.

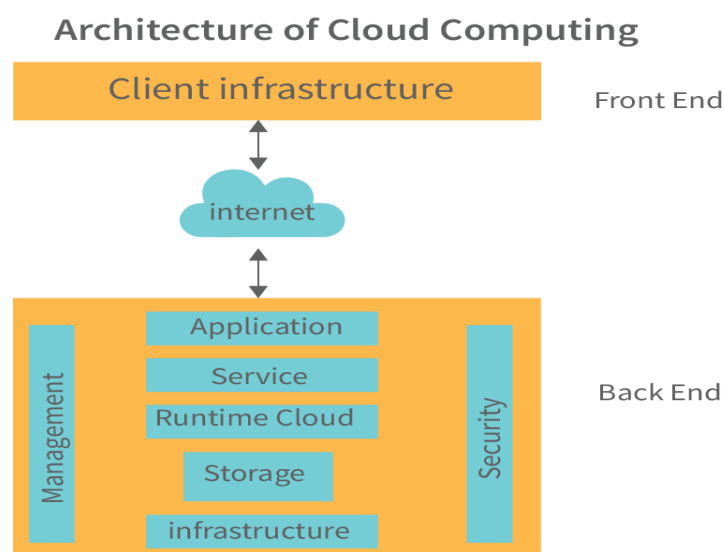


Figure 1. Architectures of Cloud Computing

Cloud computing exhibits several key characteristics that differentiate it from traditional IT environments:

- On-demand self-service: Consumers can autonomously provision computing resources, such as server time and network storage, without requiring direct interaction with each service provider. This enables users to quickly obtain the resources they need without manual intervention.

- Broad network access: Cloud capabilities are accessible over the network and can be utilized through standard mechanisms, allowing for seamless usage across various client platforms, including mobile devices, tablets, laptops, and workstations.

- Resource pooling: Cloud providers consolidate their computing resources into a shared pool to serve multiple consumers in a multi-tenant model. Physical and virtual resources are dynamically assigned and reassigned based on consumer demand. While consumers may not have direct control or knowledge of the exact resource location, they can specify higher-level preferences such as the country, state, or datacenter.

- Rapid elasticity: Cloud services can quickly scale up or down to meet changing demands. Resources can be provisioned and released elastically, often automatically, allowing for rapid scaling in response to fluctuations in demand. From the consumer's perspective, the cloud appears to offer unlimited resources that can be allocated in any quantity and at any time.

- Measured service: Cloud systems have built-in mechanisms to monitor, control, and optimize resource utilization. Metering capabilities enable the measurement of resource usage, such as storage, processing power, bandwidth, and active user accounts. This measurement facilitates transparent and granular billing, often on a pay-per-use or charge-per-use basis. Both providers and consumers have visibility into the resources being utilized and their associated costs.

These characteristics collectively contribute to the flexibility, scalability, and cost efficiency of cloud computing, providing users with on-demand access to a wide range of computing resources while optimizing resource allocation and consumption.

Reference:

1. N. Carr, *The Big Switch: Rewiring the World, from Edison to Google*. W. W. Norton & Co., New York, 2020.

2. The Research and Application of Network Teaching Platform Based on Cloud Computing, Z. Tao and J. Long, *International Journal of Information and Education Technology*, Vol. 1, No. 3, August 2011

WHAT CAN YOU CREATE WITH THE INSTAGRAM API. RESTRICTIONS VS CAPABILITIES. AI ADOPTION

Petruk Viktoriia

Kharkiv National Automobile and Highway University, Kharkiv

Over time, Instagram made a lot of restrictions on using their API.

The most significant of them is getting information about accounts without logging from public accounts if they are not in business mode, another one is restrictions with reading and replying to your messages, even if you have the token and key, Instagram API is not allowing you to do this.

In the article, we can see how we can use IG API effectively and bypass some restrictions.



First question off the top of my head, is it legal to collect data from Instagram?

Many created apps for collecting data from Instagram that somehow wanted to replace it were blocked by Instagram itself. Depending on IG's terms of service for protecting user privacy we can do it **only with official APIs**, and then we can predict that Web scraping is not a good way to collect and analyze some data from IG or Facebook. Then what can we do and create with Instagram API as it is?

Let`s dive deep with some examples.

1 Case : Imagine we want to create an SMM tool that will analyze our previous content and their activity, and propose topics and content plans for the next posts.

2 Case : We want to extend our implementation. Can we analyze some number of posts of similar businesses and depending on their statistics define our future “more successful” content plan?

3 Case : Let`s raise the stakes. Can we create a personal assistant who will propose posts for us, publish posts, and read and answer comments and messages outside of IG?

Let`s investigate the facilities of the Instagram API.

1. Instagram Graph API and Business Discovery

This API allows us to interact with business and professional accounts. By using this API you can get a certain count of the comments, likes, even they were hidden in that account, followers, pictures, and even videos of the posts.


The basic request will be:

```
GET /{ig-user-id}/business_discovery
```

And the answer:

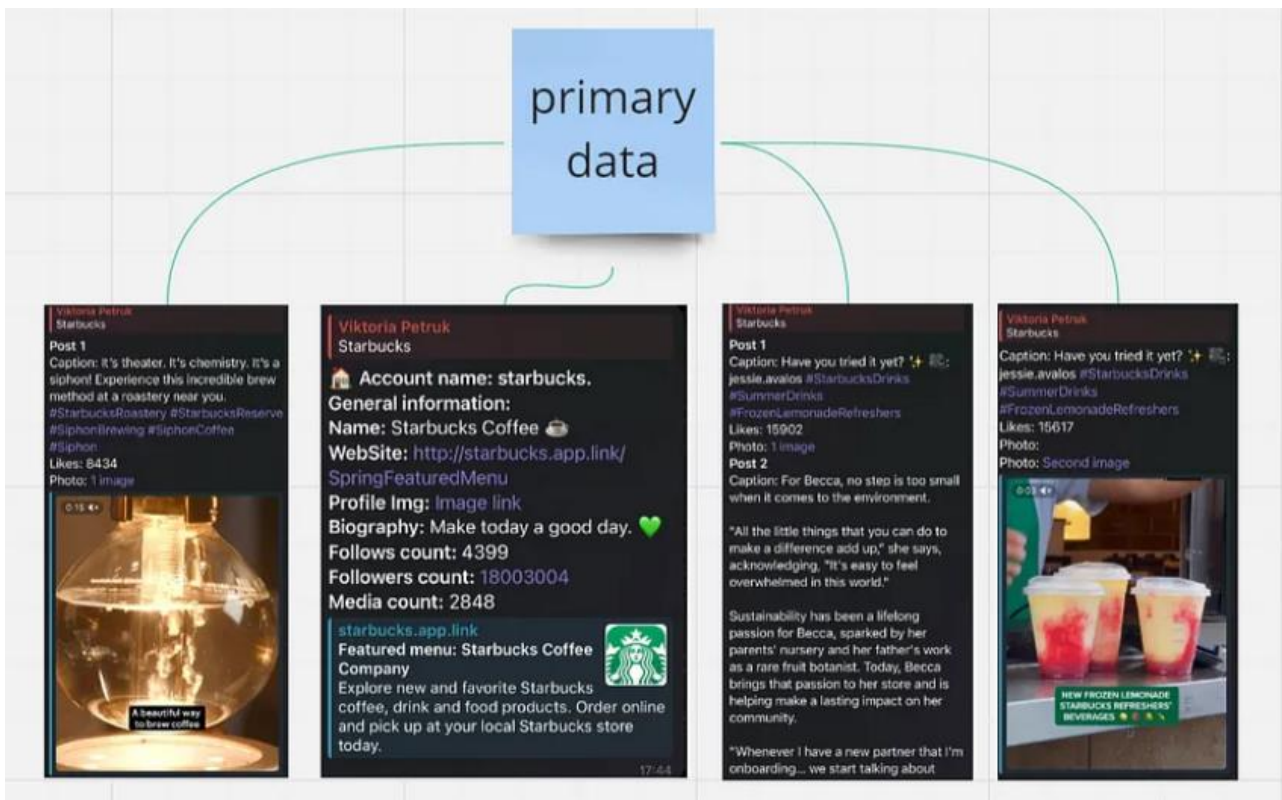
```
{
  "business_discovery": {
    "followers_count": 267793,
    "media_count": 1205,
    "media": {
      "data": [
        {
          "comments_count": 50,
          "like_count": 5841,
          "id": "17858843269216389"
        },
        {
          "comments_count": 11,
          "like_count": 2998,
          "id": "17894036119131554"
        },
        {
          "comments_count": 28,
          "like_count": 3644,
          "id": "17894449363137701"
        }
      ],
      "id": "17841401441775531"
    },
    "id": "17841405976406927"
  }
}
```

To start using it you need to follow these steps:

1. Create your developer account *t* in Meta for developers.
2. Create a business IG account and link it to the Facebook.
3. Get a key  of your Instagram user, using <https://developers.facebook.com/tools/explorer/>.
4. Add special permissions: *instagram_basic*, *instagram_manage_insights*, *pages_show_list*.
5. Generate a token with these permissions for your application.

For more details you can reach out the documentation.

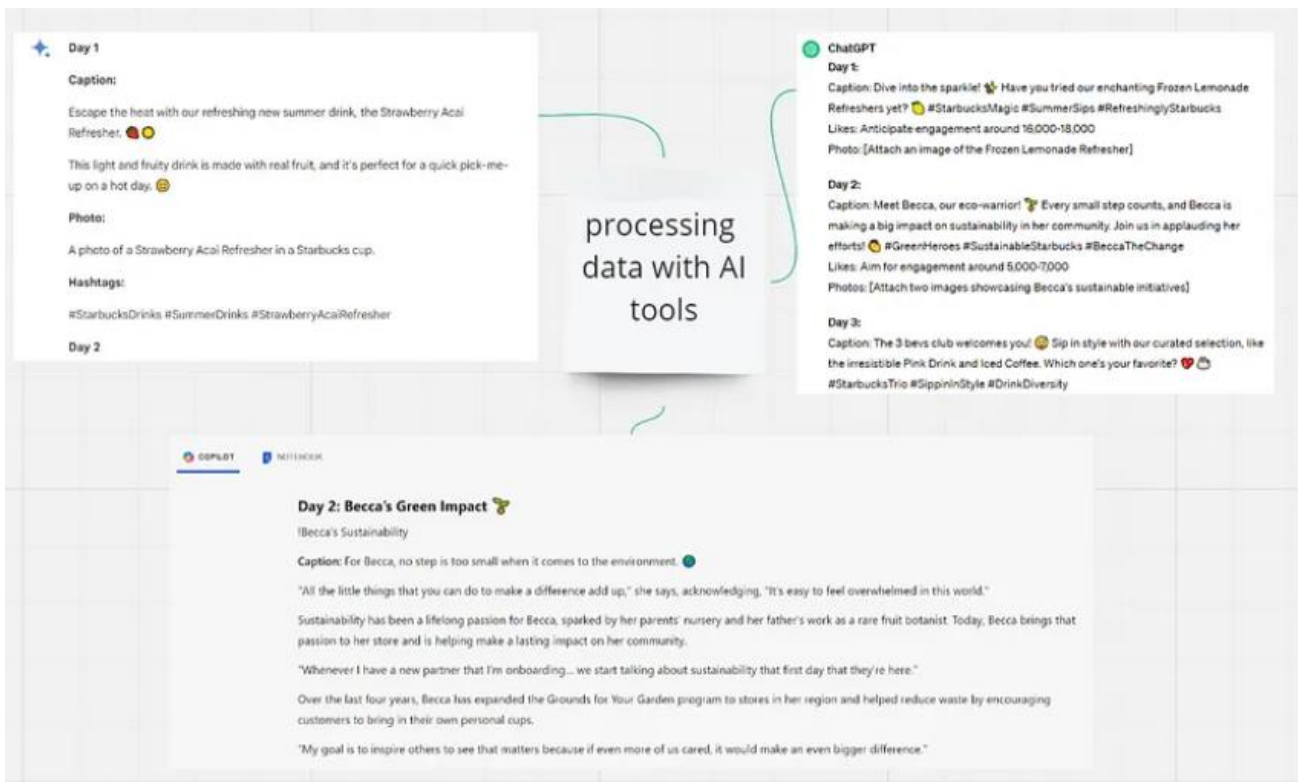
Grab our primary data from Instagram to process it with AI to generate our future content plan:



The data was obtained through the Instagram API with the combining API of the chat messenger.

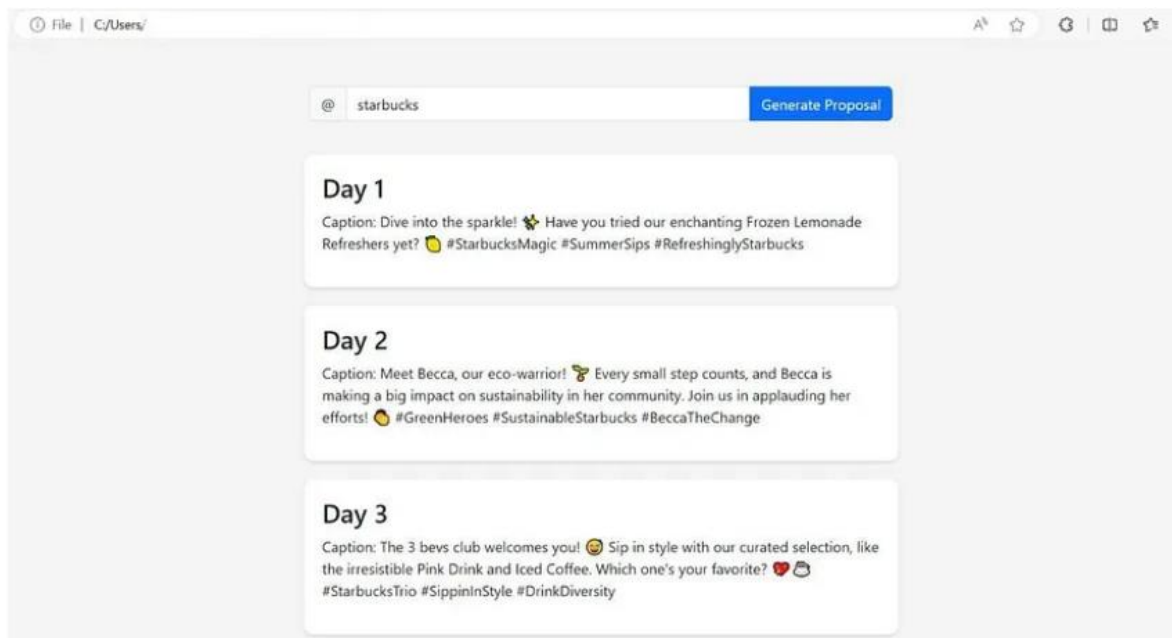
The next step: processing them. Which AI tools we can use to analyze these posts?

- I tried ChatGpt, Gemini, and Copilot. Let`s verify how they can process our data and ask them to create a content plan based on getting data from Instagram:



We can see different results and they are quite good, in the future, we can implement the API for the interaction with one of the presented services.

Based on this now then build the first version of our posts generator. Create a simple web page and connect the API:



On the web page we use the input field to get the name of the business account, then using Instagram API we request to get the latest posts: `_httpClient.GetAsync($"/{BusinessAccountID}")`

```
+ "?fields=business_discovery.username({username})"+
+ "{ {username,website,name,ig_id,id,profile_picture_url,biography,follows_count,followers_count,media_count,"+
+ "media { {id,like_count,media_type,media_url,caption} } } }"+
+ "&access_token={AccessToken}");
```

Then we process these posts with proposal AI APIs and show the result of the generated content plan on the page.

Conclusion: by using AI and Instagram API we can do our 1 case and partly 2 case and extend it with different functionality.

If you need to get basic information, you can use Instagram Basic Display API, it can get you persimmons to any of the accounts, but with only basic data.

Also, you can investigate how to get content by using hashtags: [Hash Tag Search Instagram](#).

Moving on to the second case, can we get a static of the different accounts?
2. API Graph Statistic □

Statics are only available for our professional account and we can't get them for other public accounts, but using Business Discovery, we can create our own statics!

For example, after loading the last 50–100 posts, and analyzing the count of likes, and comments, the count of followers in a certain moment.

! Restriction: you can't get personal statistics if you have less than 100 followers.

Overall, the answer to the 2d case: we can implement it with the facilities of the Instagram API, but to prove our customer's analytics we need to develop our tool based on getting parameters.
 Basic example of the web-page with statistics:



3. Graph API Content Publishing

Our subsequent case — creating a personal Instagram assistant, purpose is publishing different content, images, and videos, and setting geolocation for the media.

Abilities of the IG API:

- Publishing media:

With this API you can add posts to your professional account, to achieve this you need to have your userID and add permissions for generating the token in Graph Explorer.

```
Permissions: ads_management, business_management, instagram_basic,
instagram_content_publish, pages_read_engagement.
```

Basic POST request:

```
POST /{ig-user-id}/media_publish
```

Response:

```
{ "id": "17920238422030506" // IG Media ID }
```

Setting geolocation:

For the setting geolocation directly you need to have location_id, which you can get with the additional request:

```
{ "method": "GET", "url":
"https://graph.instagram.com/v12.0/locations/search", "params":
{ "access_token": "YOUR_ACCESS_TOKEN",
  "lat": 40.7128,
  "lng": -74.0060,
  "distance": 1000
}
}
```

Also, you need to set here latitude and longitude.

After getting the location_id, you can make a POST request for publication including this parameter.

Replies to comments:

1. Fetch the comments associated with a particular post using the API. You can use the `/{media-id}/comments` endpoint to retrieve comments for a specific media item.

2. Then, identify the Comment to Reply to: Once you have the comments, identify the comment you want to reply to.

3. Craft Your Reply: Prepare the reply message you want to send.

4. Post Your Reply: Use the endpoint to post your reply to the selected comment:

```
POST /{ig-comment-id}/replies
```


- Publishing stories and reels:

! Restrictions:

- 1) We can't get stories of other accounts in the current API implementation, only stories of our accounts with GET `/ig-user-id/stories`.
- 2) Currently, only business accounts can publish stories using the Content Publishing API.

For publishing stories, you need to use the same request as for media but add `media_type=STORIES`, and specify the path to the image or video in the `image_url` or `video_url` parameter.

For publishing reels, you need to add `media_type=REELS`.

About **user tags**, **product tags**, **collaborator tags**, and **creating a Carousel Container**, you can read more in the official documentation: <https://developers.facebook.com/docs/instagram-api/guides/content-publishing>.

Proceeding with all information, we assume that with some restrictions, we can create a personal assistant for us, but we have restrictions on interaction with other accounts (seeing their stories, putting likes).

The last calling question, what about messages? Can we chat with someone out of the Instagram application border?

In primary Instagram API, we can not do this, will investigate in the next article how we can cope with this by using Facebook Messenger API.

УДК 004.774

ВДОСКОНАЛЕННЯ СТВОРЕННЯ ФОРМ НА БАЗІ БІБЛІОТЕЦІ REACTJS

*Гриценко К.М., студент магістратури механічної кафедри
Харківського національного автомобільно-дорожнього університету*

Анотація: У статті розглянута актуальна проблема та необхідність вдосконалення роботи з динамічними формами на базі бібліотеці ReactJS, та як вдосконалити та зменшити час необхідний для розробки форм управління даними.

Ключові слова: ReactJS, форми, оптимізація, динамічне керування полями, залежності, шаблонні залежності.

Актуальність: На теперішній час ReactJS є широко використовуваною JavaScript-бібліотекою для створення клієнтських інтерфейсів. Особливо акту-

альним є його використання при розробці форм для введення та редагування даних, оскільки це одне з основних завдань у багатьох веб-додатках. При цьому неефективна розробка таких форм може призвести до погіршення досвіду користувача, зниження продуктивності додатків і навіть до вразливостей безпеки. Тому оптимізація форм розробки в ReactJS стає важливою для забезпечення високого рівня продуктивності, безпеки та зручності використання. Ключові аспекти оптимізації включають контрольовані компоненти, скорочення кількості перемальовок, обробку подій та перевірку введення даних, а також забезпечення доступу до даних лише через спеціальні функції доступу (акцесори). Ці заходи допомагають розробникам підвищити продуктивність своїх програм, надаючи більш стандартну, зручну та безпечну форму для роботи з даними.

Існує багато пакетів NPM для роботи з формами в ReactJS, які можуть значно облегшити процес розробки та підвищити ефективність кодування.

Наприклад, Formik - це популярний пакет, який пропонує спрощений спосіб управління формами в React. Він надає готові НОС і функціональні компоненти, які автоматизують управління станом форми, валідацію і відправку даних. Крім того, він підтримує інтеграцію з такими популярними бібліотеками, як Yup для перевірки форм і Redux для управління станом додатків.

Другий відомий пакет - це Final Form. Він також призначений для роботи з формами в React і пропонує подібний набір можливостей, як і Formik. Однак Final Form має кілька відмінностей, наприклад, він краще працює з великими формами завдяки своїй архітектурі, яка мінімізує кількість перерендерінгу.

React Hook Form - ще один популярний пакет, який дозволяє використовувати хуки React для роботи з формами. Він простий в освоєнні та пропонує легкий синтаксис для роботи з формами. Крім того, він має малий об'єм і не вимагає додаткових залежностей.

Усі три пакети мають активний розвиток і підтримують велику спільноту розробників, що робить їх хорошим вибором для роботи з формами в ReactJS. Вибір залежить від конкретних вимог проекту та особистих переваг розробника.

Дослідивши наявні пакети для роботи з формами було виявлено що немає пакета задля керування полями форми у залежності від введених даних.

Для роботи з даними часто потрібно керувати полями форми динамічно, наприклад:

- сховати поле якщо значення якогось поля у формі прийняло/не прийняло таке конкретне значення;
- показати поле якщо якесь поле у формі прийняло/не прийняло таке конкретне значення;
- дозволити/заборонити редагування поля у формі прийняло/не прийняло таке конкретне значення;

- встановити значення або очистити одного або багатьох полів, якщо інше поле у формі прийняло/не прийняло таке конкретне значення.

Перелічені завдання часто зустрічаються при створенні форм для роботи з даними.

Давайте розглянемо як можна реалізувати ці завдання, використовуючи популярні пакети для роботи з формами.

Оскільки пакети взяті за приклад однаково вирішують перелічені завдання, розглядатимемо лише один пакет, наприклад візьмемо пакет Formik.

Розглянемо конкретну імплементацію залежного поля з інших полів (рис. 1):

```
const MyField = (props) => {
  const {
    values: { textA, textB },
    touched,
    setFieldValue,
  } = useFormikContext();
  const [field, meta] = useField(props);

  React.useEffect(() => {
    // set the value of textC, based on textA and textB
    if (
      textA.trim() !== '' &&
      textB.trim() !== '' &&
      touched.textA &&
      touched.textB
    ) {
      setFieldValue(props.name, `textA: ${textA}, textB: ${textB}`);
    }
  }, [textB, textA, touched.textA, touched.textB, setFieldValue, props.name]);

  return (
    <>
      <input {...props} {...field} />
      {!!meta.touched && !!meta.error && <div>{meta.error}</div>}
    </>
  );
};
```

Рисунок 1 – Код, що реалізує залежність одного поля від двох інших

Таким чином реалізація залежного поля в кожному індивідуальному випадку описується кодом, що призводить до роздування коду та складності управління шаблонними залежностями, також при зміні будь-якої логіки або значень, необхідно залучати розробника для того, щоб змінити логіку роботи кожного окремого поля, що призводить до інкрементального ускладнення проекту та збільшення його вартості як у розробці так і за подальшої підтримки.

Метою мого дослідження на майбутнє є спрощення управління залежностями у формах за рахунок розробки шару абстракції роботи із залежностями та зручне управління поданням форми в іншому вигляді.

Висновок: реалізація динамічних залежностей між полями форми може призводити до ускладнення коду та збільшення часу на розробку та технічну підтримку. Проблема ускладнюється тим, що будь-яке редагування в логіці або параметрах вимагає втручання розробника для коригування коду кожного окремого поля. Метою майбутніх досліджень є розробка засобів, що спрощують управління залежностями у формах та надають більш зручний спосіб подання форми. Рекомендую використовувати популярні пакети, такі як Formik, React Hook Form або Final Form, щоб звести до мінімуму час на розробку та підтримку коду.

Література

1. ReactJs: Бібліотека для веб-інтерфейсу та нативного інтерфейсу користувача [Електронний ресурс]. Режим доступу: <https://react.dev/>. Дата доступу: 15.04.2024
2. Formik: world's most popular open source form library for React and React Native. [Електронний ресурс]. Режим доступу: <https://formik.org/>. Дата доступу: 15.04.2024
3. Final Form: Framework agnostic, high performance, subscription-based form state management. [Електронний ресурс]. Режим доступу: <https://final-form.org/>. Дата доступу: 15.04.2024
4. React Hook Form: Performant, flexible and extensible forms with easy-to-use validation. [Електронний ресурс]. Режим доступу: <https://react-hook-form.com/>. Дата доступу: 15.04.2024

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ C-V2X ТА DSRC НА АВТОНОМНОМУ ТРАНСПОРТІ

Карпішен Б.С.

Харківський національний автомобільно дорожній університет, м. Харків

Анотація: Інформаційно комунікаційні технології відіграють важливу роль в сучасному автономному водінні. Робота порівнює характеристики технологій C-V2X та DSRC беручи до уваги результати реального тестування систем. У висновку порівнюються характеристики та розкриваються окремі сфери використання технологій.

Ключові слова: Інформаційно комунікаційні системи, V2V, C-V2X, DSRC, тестування.

Інформаційно-комунікаційні системи на транспорті відіграють важливу роль у розвитку сучасних транспортних мереж. Ці системи забезпечують ефективний зв'язок між транспортними засобами, інфраструктурою та користувачами, підвищуючи безпеку, ефективність і комфорт дорожнього транспорту.

Інформаційно-комунікаційні системи (ІКС) – це комплексні технології, які забезпечують обмін інформацією між транспортними засобами та інфраструктурою, що включає мережі зв'язку, датчики, камери, системи керування та аналіз даних. ІКС є важливим компонентом в автономному водінні. Вони забезпечують автомобілям можливість спілкуватися між собою (V2V), з інфраструктурою (V2I) та іншими об'єктами (V2X). Вони мають повну технічну правову та законодавчу базу, але технологія ще досі знаходиться більше на етапах розробки ніж впровадження.

Основними представниками виступають дві технології комунікації C-V2X та DSRC.

C-V2X (Cellular Vehicle-to-Everything) – це технологія, яка використовує стільниковий зв'язок для обміну інформацією між транспортними засобами та інфраструктурою. Ця технологія забезпечує високу надійність зв'язку і широкий спектр можливостей для обміну даними, таких як попередження про небезпечні ситуації, моніторинг трафіку, оптимізація маршрутів та інше [1].

C-V2X підтримується багатьма мобільними операторами, провідними виробниками мобільного обладнання та автовиробниками. C-V2X пропонує два режими зв'язку – прямий і мережевий. Прямий зв'язок між транспортними засобами, інфраструктурою та іншими учасниками дорожнього руху за безпечується за допомогою спектру інтелектуальної транспортної системи (ІКС), який був визначений і гармонізований на міжнародному рівні з метою безпеки. Це працює незалежно від стільникових мереж і є основою для стандартизованих глобальних систем безпеки.

Мережевий зв'язок, у якому C-V2X використовує мобільну мережу для отримання інформації, дає змогу платформам C-V2X поєднувати захищені широкі зони з'єднання невеликого радіусу дії в одному модулі та відкриває нові привабливі випадки використання від допомоги при водінні до більш дорогих інформаційно-розважальних засобів. C-V2X розроблено для повної сумісності з мобільними технологіями 5G, а можливості технології розширені завдяки впровадженню стандартів 3GPP [2].

В епоху повсюдного використання 5G C-V2X зможе підтримувати низку розширених служб безпеки, включаючи точне позиціонування та визначення дальності, щоб уможливити спільне та автоматизоване водіння, надання локальних динамічних карт на основі даних камери та датчиків, а також підключення з низькою затримкою, необхідне для забезпечення високої щільності технології взводу (platooning).

Як і 802.11p, C-V2X використовує глобальну навігаційну супутникову систему (GNSS) для визначення положення транспортного засобу та синхронізації зв'язку між транспортними засобами та з придорожною інфраструктурою. У цьому режимі SIM-карта не потрібна, оскільки автомобіль не потребує підключення до стільникової мережі. Транспортний засіб та його водій залишаються анонімними, оскільки для прямого зв'язку безпеки не потрібна підписка на стільниковий зв'язок.

Стільниковий V2X (C-V2X) зазвичай використовує для зв'язку смугу частот 5,9 ГГц – це офіційно призначена частота інтелектуальної транспортної системи (ITS) у більшості країн. Використовуючи діапазон, технологія може підтримувати прямі з'єднання з низькою затримкою на короткі відстані без залучення стільникової мережі.

C-V2X може функціонувати без допомоги мережі та має дальність зв'язку, що перевищує 1,6 км висхідного зв'язку 200 Мбіт/с для LTE-A [2].

DSRC (Dedicated Short-Range Communications) – це технологія короткострокового зв'язку, яка працює в діапазоні частот 5.855–5.925 ГГц та може відрізнитись відповідно регіональним стандартам. Вона використовується для зв'язку між транспортними засобами та інфраструктурою на коротких відстанях. DSRC дозволяє обмінюватися інформацією про швидкість, місцезнаходження та інші параметри в режимі реального часу.

Таким чином, 5,9 ГГц зв'язок на короткій відстані (DSRC) базується на стандарті IEEE 802.11p, який став перспективним рішенням, і реалізований у транспортному засобі транспортного засобу (V2V), автомобільному транспорті (V2R) і транспортному засобі до інфраструктури (V2I), а також у розробці автономних транспортних засобів. DSRC використовує мультиплексування з ортогональним частотним поділом (OFDM), іноді на основі стандарту бездротової локальної мережі (WLAN), відомого як стандарт IEEE 802.11a. У прямому порівнянні з IEEE 802.11a, DSRC має надійну стійкість до перешкод і може адаптуватися до безперервних і швидких умов переходу. Структурування каналу DSRC передбачає сім каналів з частотою 10 МГц, з дальністю ефективної

роботи на стандартній частоті до 1000 м. 802.11p представлено як бездротова технологія, здатна задовольнити вимоги до низької затримки нижче 100 мс у разі передачі інформації та повідомлень безпеки дорожнього руху [3].

Він також може адаптуватися до швидкостей 350 км/год або навіть 500 км/год залежно від діапазону частот, хоча в цих випадках погіршення продуктивності на високих швидкостях неминуче. DSRC оптимізований для "блокування прямої видимості" (нелінійний приціл) та роботи в екстремальних погодних умовах.

Таким чином, у контексті запобігання зіткненням даних DSRC покладається на доступ до кількох несучих у виявленні зіткнень (CSMA/CA). Будучи типом бездротового зв'язку з короткою та середньою дальністю, він добре підходить до концепцій систем зіткнення даних для безпеки дорожнього руху. Остаточний підхід аналізує синхронізацію та розподіл слотів, призначених для розповсюдження в мережах з багатьма переходами, з підвищеними умовами мобільності та правильно оптимізує зв'язок IEEE 802.11 MAC у підході систем, призначених для дорожнього сектора.

Стандарт IEEE802.11p WAVE є лише частиною групи стандартів, що стосуються всіх рівнів протоколів для операцій на основі DSRC. Стандарт IEEE802.11p обмежений сферою застосування IEEE802.11, яка є стандартом рівня MAC та PHY для роботи в межах одного логічного каналу IEEE802.11. Це важлива відмінність порівняно з технологіями C-V2X, оскільки периметр їх специфікації охоплює всі рівні протоколу, навіть у режимі прямого зв'язку. Усі знання та складності, пов'язані з планом каналу DSRC та робочою концепцією, охоплюються стандартами верхнього рівня IEEE1609.2 [4].

Хоча C-V2X і 802.11p можуть співіснувати в спектрі ІКС використовуючи різні канали в діапазоні 5,9 ГГц. Лише 10 МГц спектру в діапазоні 5,9 ГГц потрібно для підтримки базових служб безпеки, тоді як 70 МГц можуть підтримувати розширені служби безпеки, такі як обмін великими обсягами даних, зібраних датчиками автомобіля [5].

Обидві технології були протиставлені один одному в рамках декількох польових тестувань [6,7,8]. У якості основних характеристик порівняння були вибрані загальна затримка передачі/підключення, ефективна дальність роботи та надійність підключення.

Кожен оснащений пристроями DSRC транспортний засіб передає свою інформацію про стан, включаючи місцезнаходження, швидкість, та прискорення, кілька разів на секунду на відстані кілька сотень метрів. Кожен транспортний засіб також отримує ці повідомлення щодо безпеки.

C-V2X розподілена система планування пакетів, яка працює в синхронному режимі. "Бюджет затримки пакетів" або PDB – це вікно часу, протягом якого пакетам із потоку SPS призначаються ресурси, коли вони заплановані вперше. PDB визначає час затримки пакетів із певного потоку напівстійкого планування (SPS). Всі наступні повідомлення з одного потоку передаються точно з інтервалом періодичності повідомлень (наприклад, 100 мс між

повідомленнями). Середня та максимальна затримка залишаються незмінними, навіть якщо навантаження на систему збільшується. DSRC покладається на CSMA/CA для доступу до каналу. Планування не задіяне, і передача заснована на зондуванні енергії на каналі. Коли система злегка завантажена, повідомлення можуть передаватися з низькою затримкою. Однак, оскільки система сильно завантажується, затримка повідомлень швидко зростатиме[9]. З високою перевантаженістю затримка, а також інтервал між наступними повідомленнями значно збільшується.

Configuration	DSRC	C-V2X (PC5 Mode 4)
Channel	Channel 172	5860 MHz (Channel 172)
Bandwidth	10 MHz	10 MHz
Modulation	QPSK ½ (6 Mbps burst rate)	MCS 5
Application Used	Savari	Savari
Tx/Rx Configuration	1 Tx 2 Rx	1 Tx 2 Rx
Device Details	Savari MW1000	Qualcomm Roadrunner platform
Blind HARQ	NA	Enabled
Tx Power	21 dBm	21 dBm
Packet Size	193 Bytes	193 Bytes (5 Sub-Channels)*

* Sub-Channel size = 5 RB

Рис. 1 – Характеристики тестового обладнання [6]

Результати випробувань показують, що за відмінних умов радіозв'язку (-50 дБм Отримана потужність без додаткового шуму) обидві технології V2X можуть надійно передавати корисне навантаження BSM. У ненавантажених умовах затримка C-V2X зазвичай становить від 1 до 4 мс затримки DSRC, що з точки зору всієї системи автомобіля є незначною різницею.

Тестування дальності визначає відстань, на якій PRR або надійність прийому повідомлень BSM падають нижче прийнятного рівня. Поріг PRR для визначення діапазону становить 90%. Основні показники продуктивності включають: коефіцієнт прийому пакетів (PRR) та міжпакетний розрив (IPG) [6].

Порівняно дві технології V2X, використовуючи діапазон, надійність та IPG як KPI. У всіх тестах C-V2X OBU значно перевершували OBU DSRC. Результати випробувань свідчать про вигреш у діапазоні радіочастот для Cellular-V2X порівняно з DSRC.

Обидві технології V2X піддавались одній і тій же IEEE 802.11ac 80 МГц синтетично створеній перешкоді в рамках тестування надійності сигналу. Подібним чином, у випробуваннях на перешкоди сусіднього каналу перешкоди як для DSRC, так і для C-V2X є однаковими синтетично створеними перешкодами IEEE 802.11p 10 МГц у сусідньому каналі.

Таблиця 1 – Порівняння дальності зв'язку DSRC та C-V2X при ефективній потужності передавача 5 та 11 дБм

Процедура випробування	Дальність зв'язку, м (при 90% надійності)			
	DSRC 5 дБм	C-V2X 5 дБм	DSRC 11 дБм	C-V2X 11 дБм
Діапазон прямої видимості (LOS)	625	1050	925	1350
Нелінійне блокування зору (NLOS) (5GAA)	250/350	450	425	625
Блокування нелінійного зору (NLOS) (CAMP)	175/250	550	400	600
Співіснування з Wi-Fi (смуга пропускання 80 МГц)	–	–	550	950
Співіснування V2X із сусідньою носійною DSRC	–	–	325	950

Результати показують, що C-V2X набагато надійніший за DSRC в обох сценаріях. Посилання DSRC не працює в обох випадках. Це пов'язано або зі спотворенням отриманого сигналу, або з голодуванням CSMA/CA. Для C-V2X, коли передавач чує перешкоди, він намагається уникнути частотних перешкод. Це призводить до того, що зв'язок не впливає на більшість конфігурацій, за винятком випадків, коли смуга пропускання перешкод стає настільки широкою, що її розташування унеможлиблює повне уникнення. Якщо перешкоди налаштовані так, щоб вони були приховані від передавача, передача не може уникнути частотних перешкод [7].

У висновку технології C-V2X та DSRC відіграють ключову роль у розвитку автономного водіння, створюючи нові можливості для безпечного та надійного руху на дорогах. Інформаційно-комунікаційні системи на транспорті, зокрема C-V2X і DSRC, є ключовими технологіями для розвитку автономного водіння та підвищення безпеки на дорогах. Вони дозволяють транспортним засобам ефективно взаємодіяти з навколишнім середовищем, забезпечуючи більш безпечний і зручний дорожній рух. Обидві технології демонстрували подібні наскрізні затримки рівня додатків за умов, що не перевантажені, і обидві технології відповідали вимогам до затримки для додатків безпеки V2V, зазначених стандартами. Продуктивність міжпакетного розриву була в межах 10 мс для обох технологій V2X, як правило, дуже швидко зростала, коли пристрої виходили за межі діапазону. Однак C-V2X переважає DSRC в таких показниках як дальність зв'язку та надійності сигналу у випадку замулення каналу, що надає пріоритет використання першої технології не виключаючи використання DSRC.

Важливим рішенням, що стосується рівнів, є подібність послуг між C-V2X та DSRC. Основний набір повідомлень про безпеку є основною програмою для обох технологій прямого режиму зв'язку. Усі системи безпроводного зв'язку, страждають від недоліків, що їм властиві. Недоліки полягають у обмеженій потужності в різних сферах:

- обмежені канали, ця межа торкнеться особливо великих міст;
- обмежена швидкість передавання даних, враховуючи, що лише один автономний автомобіль буде використовувати до 4 000 ГБ даних на день [6];
- безпроводний зв'язок сприйнятливий до зовнішніх впливів, що може теж стати недоліком;
- обмеження розповсюдження даних через оточення, такі як будівлі, тунелі, а також доплерівські ефекти, що спричиняють зменшення швидкості розповсюдження за рахунок повторного передавання даних;
- витрати на забезпечення всебічного належного такої мережі, як LTE або 5G, вимагають значних фінансів.

Розвиток інформаційно-комунікаційних систем на транспорті відкриває шлях до створення більш розумних та інтелектуальних транспортних мереж. Завдяки обміну інформацією в режимі реального часу між транспортними засобами, пішоходами та інфраструктурою, ці системи сприяють зниженню кількості дорожніх інцидентів, зменшенню заторів та покращенню загальної ефективності руху.

Список літератури

1. R, Dhinesh Kumar, and Rammohan A. 2023. "Revolutionizing Intelligent Transportation Systems with Cellular Vehicle-To-Everything (C-V2X) Technology: Current Trends, Use Cases, Emerging Technologies, Standardization Bodies, Industry Analytics and Future Directions." *Vehicular Communications* 43 (October): 100638. <https://doi.org/10.1016/j.vehcom.2023.100638>.
2. Zadobrischi, Eduard, Mihai Dimian, and Mihai Negru. 2021. "The Utility of DSRC and V2X in Road Safety Applications and Intelligent Parking: Similarities, Differences, and the Future of Vehicular Communication." *Sensors* 21 (21): 7237. <https://doi.org/10.3390/s21217237>.
3. "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Certificate Management Interfaces for End Entities," in *IEEE Std 1609.2.1-2022 (Revision of IEEE Std 1609.2.1-2020)*, vol., no., pp.1-261, 30 June 2022, doi: 10.1109/IEEESTD.2022.9810154.
4. Calabrese, Michael, and Amir Nasr. 2020. "The 5.9 GHz Band Removing the Roadblock to Gigabit Wi-Fi." *New America*, July, 36. [newamerica.org/oti/reports/59-ghz-band/](https://www.newamerica.org/oti/reports/59-ghz-band/).
5. Joseph, Lentin, and Amit Kumar Mondal. 2022. *Autonomous Driving and Advanced Driver-Assistance Systems (ADAS) : Applications, Development, Legal Issues, and Testing*. Boca Raton: CRC Press/Taylor and Francis Group.
6. 5GAA. 2019. "V2X Functional and Performance Test Report; Test Procedures and Results." *V2X Functional and Performance Test Report*. 5GAA Automotive Association. https://5gaa.org/content/uploads/2018/11/5GAA_P-190033_V2X-Functional-and-Performance-Test-Report_final-1.pdf.
7. A. Rayamajhi, A. Yoseph, A. Balse, Z. Huang, E. M. Leslie and V. Fessmann, "Preliminary Performance Baseline Testing for Dedicated Short-Range Communication (DSRC) and Cellular Vehicle-to-Everything (C-V2X)," 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), Victoria, BC, Canada, 2020, pp. 1-5, doi: 10.1109/VTC2020-Fall49728.2020.9348708.
8. Zhong, Ziyi, et al. FIELD TESTS on DSRC and C-V2X RANGE of RECEPTION on UTAH ROADWAYS. Utah Department of Transportation by Panasonic Corporation of North America, 23 Apr. 2021.
9. Alsmirat, Mohammad & Al-Rifai, Saleh & Sababha, Belal. (2015). Reducing Message Loss in DSRC Networks using Dynamic Distribution of Safety Messages over EDCA Access Categories.

РОЗРОБКА ТЕЛЕГРАМ БОТА ДЛЯ ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ

Сірий Д.О.

Харківський національний автомобільно-дорожній університет, Харків

Організація пасажирських перевезень є важливою складовою транспортної системи будь-якої країни. В умовах сучасного міського життя, зростання чисельності населення та збільшення обсягів пасажирських перевезень, ефективне управління заявками на перевезення стає критично важливим. Це вимагає впровадження інноваційних технологій, які здатні автоматизувати процеси обробки заявок, забезпечити високий рівень обслуговування пасажирів та оптимізувати використання транспортних ресурсів.

Одним із таких інноваційних рішень є розробка програми для формування заявок на рейси пасажирського транспорту у вигляді Telegram-бота. Використання Telegram-бота дозволяє автоматизувати взаємодію між пасажиром та службою перевезень, забезпечуючи зручний та швидкий спосіб подачі заявок, отримання інформації про маршрути та рейси, а також керування існуючими заявками. Основною метою цієї курсової роботи є розробка Telegram-бота для організації пасажирських перевезень з використанням принципів об'єктно-орієнтованого програмування (ООП).

Аналіз предметної галузі організації пасажирських перевезень

Загальна характеристика пасажирських перевезень

Пасажирські перевезення є важливою складовою транспортної системи будь-якої країни. Вони забезпечують мобільність населення, сприяють розвитку економіки та підвищенню якості життя. Умовах зростаючого урбанізму та підвищення мобільності населення ефективна організація пасажирських перевезень стає критично важливою.

Значення пасажирських перевезень:

1. **Забезпечення мобільності населення:** Транспортні засоби надають людям можливість пересуватися з одного місця в інше, що є необхідним для здійснення повсякденних справ, роботи, навчання, відпочинку та інших активностей.

2. **Підтримка економічної діяльності:** Ефективна транспортна система сприяє розвитку бізнесу, оскільки забезпечує швидке та зручне пересування працівників, клієнтів та товарів.

3. **Соціальна інтеграція:** Транспорт дозволяє людям підтримувати соціальні зв'язки, що є важливим для збереження соціальної гармонії та взаємодії в суспільстві.

4. **Розвиток туризму:** Добре організована система пасажирських перевезень сприяє розвитку туризму, оскільки забезпечує зручний доступ до туристичних об'єктів та послуг.

Принципи організації пасажирських перевезень:

1. Безпека: Забезпечення безпеки пасажирів є найважливішим аспектом організації перевезень.

2. Ефективність: Ефективне використання транспортних засобів та інфраструктури дозволяє знизити витрати на перевезення, скоротити час в дорозі та підвищити задоволеність пасажирів.

3. Зручність: Пасажирські перевезення повинні бути зручними для користувачів.

4. Доступність: Транспортні послуги повинні бути доступними для всіх верств населення, включаючи людей з обмеженими можливостями.

5. Екологічність: Використання екологічно чистих видів транспорту та технологій дозволяє знизити негативний вплив на навколишнє середовище.

Функціональні вимоги до програми

Розробка програми для формування заявок на рейси пасажирського транспорту вимагає ретельного аналізу функціональних вимог. Ці вимоги визначають, які функції та можливості повинна мати програма для забезпечення ефективного і зручного обслуговування пасажирів.

Основні функціональні вимоги:

1. Управління заявками на перевезення:

- Додавання заявки: Користувачі повинні мати можливість вказувати пункт призначення, номер маршруту, прізвище та ініціали пасажирів, бажану дату поїздки, а також автоматично розраховану вартість проїзду на основі маршруту.

- Видалення заявки: Користувачі повинні мати можливість видалення раніше поданих заявок.

- Фільтрація заявок: Програма повинна надавати можливість фільтрації заявок за номером маршруту та датою поїздки.

- Перегляд всіх заявок: Користувачі повинні мати можливість переглядати список всіх поданих заявок з детальною інформацією про кожну з них.

- Очищення всіх заявок: Програма повинна забезпечувати можливість видалення всіх поданих заявок одночасно.

2. Інформація про маршрути:

- Перегляд списку маршрутів: Користувачі повинні мати можливість переглядати список доступних маршрутів з їх описом та вартістю проїзду.

- Перевірка валідності маршруту: Програма повинна мати можливість перевіряти валідність номера маршруту, вказаного користувачем при поданні заявки.

3. Інтерактивний інтерфейс користувача:

- Головне меню: Програма повинна надавати користувачам головне меню з основними функціями.

- Кнопки взаємодії: Головне меню повинно містити кнопки, які дозволяють користувачам швидко обирати потрібну дію.

- Інтерактивні повідомлення: Програма повинна використовувати інтерактивні повідомлення для інформування користувачів про результати їх дій.

4. Адміністративні функції:

- Збереження даних: Програма повинна автоматично зберігати дані про заявки та маршрути у файли для забезпечення їх збереження та можливості подальшого використання.

- Завантаження даних: Дані про заявки та маршрути повинні автоматично завантажуватися з файлів при запуску програми.

- Обробка помилок: Програма повинна забезпечувати обробку можливих помилок, таких як некоректний формат введення даних.

5. Автоматичний розрахунок вартості проїзду: Програма повинна автоматично розраховувати вартість проїзду на основі маршруту, вказаного користувачем при поданні заявки.

Програма для формування заявок на рейси пасажирського транспорту

Опис середовища розробки

Середовище розробки відіграє важливу роль у створенні ефективних та зручних програм. Для розробки Telegram-бота було використано такі інструменти та технології:

- Python: Основна мова програмування, яка забезпечує простоту і гнучкість розробки.

- Telegram API: Інтерфейс для взаємодії з платформою Telegram.

- Бібліотеки: Використання бібліотек python-telegram-bot для реалізації функціональності бота.

Розробка основних функцій бота

Основні функції бота включають:

1. Додавання заявки: Користувачі можуть подавати заявки на перевезення через бот, вказуючи необхідну інформацію.

2. Видалення заявки: Користувачі можуть видаляти раніше подані заявки.

3. Фільтрація заявок: Користувачі можуть фільтрувати заявки за номером маршруту та датою поїздки.

4. Перегляд заявок: Користувачі можуть переглядати список усіх заявок.

5. Очищення заявок: Користувачі можуть видаляти всі заявки одночасно.

Опис інтерфейсу користувача

Інтерфейс користувача Telegram-бота повинен бути зручним та інтуїтивно зрозумілим. Основні компоненти інтерфейсу включають головне меню з кнопками для вибору дій, інтерактивні повідомлення для інформування користувачів та зворотний зв'язок у випадку помилок.

Висновок

Розробка Telegram-бота для організації пасажирських перевезень дозволяє значно підвищити ефективність та зручність процесу подання заявок на перевезення. Використання принципів ООП забезпечує гнучкість та масштабованість програми, що робить її легко підтримуваною та розширюваною. Завдяки автоматизації процесів обробки заявок підвищується рівень обслуговування пасажирів, оптимізується використання транспортних ресурсів та забезпечується зручний доступ до інформації про маршрути та рейси.

Література

1. Official Documentation of Python. [Електронний ресурс]. Режим доступу: <https://docs.python.org/3/>. Дата доступу: 15.04.2024
2. Official Documentation of python-telegram-bot [Електронний ресурс]. Режим доступу: <https://python-telegram-bot.readthedocs.io/>. Дата доступу: 15.04.2024
3. Real Python [Електронний ресурс]. Режим доступу: <https://realpython.com/>. Дата доступу: 15.04.2024

УДК 004.77

РЕАЛІЗАЦІЯ ВНУТРІШНЬОЇ МАРШРУТИЗАЦІЇ МІЖ РОУТЕРАМИ З КІЛЬКОМА ПРОВАЙДЕРАМИ НА ОСНОВІ ROUTEROS

Кудінов Є.О., аспірант,

Харківський національний автомобільно-дорожній університет

Анотація. Проведено аналіз можливостей операційної системи RouterOS для використання динамічної маршрутизації локальної мережі в умовах підключення до кількох провайдерів інтернету та за наявності кількох тунелів для зв'язку між підрозділами однієї організації.

На сьогоднішній день багато організацій мають розгалужену структуру підрозділів, фізично розташованих у різних регіонах. У таких випадках одним з найважливіших завдань є завдання з'єднання всіх підрозділів у єдину, надійну та безпечну локальну мережу. Крім того, важливу роль відіграє загальна вартість цього проекту. З такої точки зору перспективними рішеннями є надійні та недорогі пристрої з великим вибором протоколів маршрутизації та віртуальних приватних мереж (VPN). Це дозволяє гнучко підлаштовувати пристрої під різні вимоги провайдерів інтернет-мереж, а також забезпечувати високу безпеку та надійність тунелів між підрозділами.

Одним із таких рішень є пристрої латвійського виробника мережевого обладнання MikroTik [1], що використовують операційну систему RouterOS. Завдяки єдиній операційній системі, функціонал усієї лінійки маршрутизаторів майже однаковий. Різниця у потужності, що дозволяє легко вибрати відповідний пристрій. При необхідності RouterOS можна встановити на персональний комп'ютер з архітектурою x86_64 або на віртуальну машину.

Для створення надійної локальної мережі між підрозділами на кожному з них має бути підключено не менше двох провайдерів інтернет зв'язку. Між собою підрозділи з'єднуються тунелями VPN. Кількість тунелів залежить від кількості провайдерів кожного з підрозділів, що з'єднуються, а також від того які саме тунелі VPN будуть використовуватися.

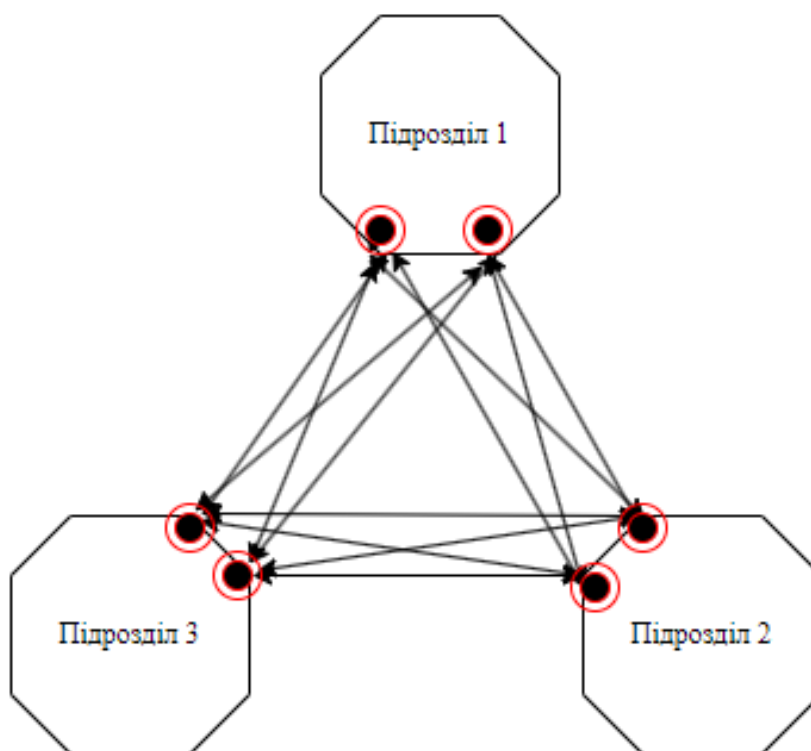


Рисунок 1 – Схема тунелів «кільце» на основі ірір + ірsec

При використанні схеми типу «кільце» та двох провайдерів у кожному підрозділі (рис.1), виходять 8 тунелів у кожному підрозділі. Це за умови застосування шифрованих тунелів ірір, яким треба вказати вихідну локальну ір-адресу (щоб пакети йшли тільки через вказаного провайдера) та віддалену адресу одержувача.

Тунелі ірір вимагають використання лише «білих» ір адрес. Найчастіше використовується схема підключення "зірка". Вона має на увазі, що всі основні послуги локальної мережі знаходяться в одному головному підрозділі (підрозділ 1) і всі інші підрозділи з'єднуються лише через нього. У такому разі, прямий зв'язок між підрозділами 2 та 3 не потрібен (рис. 1).

Якщо для тунелю використовувати протоколи типу «точка-точка», наприклад, sstp або wireguard, то кількість тунелів можна скоротити. Оскільки, на відміну ірір, ці протоколи вимагають обов'язкової вказівки віддаленої адреси лише із боку «клієнта» (рис. 2).

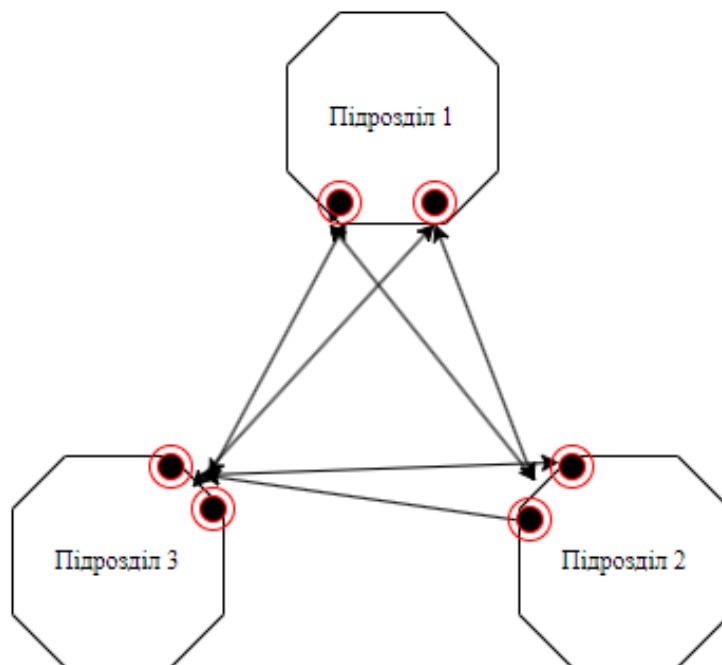


Рисунок 2 – Схема тунелів «кільце» на основі wireguard

За рахунок використання таких протоколів, не має значення, який провайдер інтернет використовується в даний момент. Тунель буде перепідключено через робочого провайдера [1].

У схем на основі ірір і wireguard є свої переваги та недоліки. Виходячи з них, слід вибрати більш перспективніший варіант підключення.

До мінусів варіанта з ірір можна віднести: більш високу складність конструкції за рахунок більшої кількості тунелів; вимога «білого» ір; на пристроях зі слабким процесором швидкість через ірір значно нижча ніж через wireguard. Наприклад, при використанні MikroTik RB2011 швидкість трафіку по шифрованому ірір тунелю становить 18-20Мб/с, а по тунелю wireguard до 40Мб/с. Втім, на потужніших пристроях ця різниця згладжується.

До переваг схеми з ірір можна віднести більш високу швидкість перемикавання маршрутів на робочий тунель за допомогою динамічної маршрутизації, за рахунок того, що підрозділи з'єднані між собою вже піднятими тунелями через кожного провайдера. У схемі з wireguard, знадобиться набагато більше часу на перепідключення, у разі відключення на «клієнті» основного провайдера та переходу на резервного.

Для маршрутизації трафіку між підрозділами через побудовані тунелі та швидкого перемикавання між ними потрібен протокол динамічної маршрутизації.

RouterOS надає три протоколи динамічної маршрутизації для внутрішніх мереж - це RIP, OSPF та IS-IS. RIP на сьогоднішній день вважається застарілим, хоча багато мережевих пристроїв досі його підтримують. OSPF та IS-IS обидва працюють на основі алгоритму Дейкстри. Основна відмінність у тому, що OSPF працює на верхньому рівні IP та зразу створювався під IP протокол, а IS-IS працює через рівень L2 і до IP не прив'язаний. Найбільш поширеним та універсальним в ір-мережах є OSPF [2].

При проектуванні мережі з кількома провайдерами та динамічною маршрутизацією, слід враховувати швидкість, якість та надійність кожного провайдера. Виходячи з цих факторів потрібно задати вартість (cost) кожному з тунелів у OSPF .

Для реалізації схеми «кільце» на основі ірір + ірsec (Рис.1) слід кожному з тунелів призначити свою підмережу. Достатньо мережі з маскою /30, з призначеною адресою цієї мережі кожному з двох маршрутизаторів в тунелі. На кожному маршрутизаторі буде 8 тунелів. Виходячи зі схеми, всі три маршрутизатори знаходяться в одній зоні. Це магістральна нульова зона. Так як у кожній такій підмережі всього дві ір-адреси для видачі їх маршрутизаторам, тип мережі в OSPF можна поставити як «точка-точка». На Рис. 3-5 наведені приклади OSPF налаштування у останньої версії RouterOS [3].

```
/routing ospf instance
add disabled=no name=default-v2 redistribute=connected,static,vpn,dhcp,modem
/routing ospf area
add area-id=0.0.0.0 comment="backbone" disabled=no instance=default-v2 name=backbone
/routing ospf interface-template
add area=backbone cost=10 disabled=no interfaces=іpip-tunnel networks=192.168.100.0/30 priority=1 type=ptp
add area=backbone cost=20 disabled=no interfaces=іpip-tunne2 networks=192.168.100.4/30 priority=1 type=ptp
add area=backbone cost=30 disabled=no interfaces=іpip-tunne3 networks=192.168.100.8/30 priority=1 type=ptp
add area=backbone cost=40 disabled=no interfaces=іpip-tunne4 networks=192.168.100.12/30 priority=1 type=ptp
add area=backbone cost=10 disabled=no interfaces=іpip-tunne5 networks=192.168.100.16/30 priority=1 type=ptp
add area=backbone cost=20 disabled=no interfaces=іpip-tunne6 networks=192.168.100.20/30 priority=1 type=ptp
add area=backbone cost=30 disabled=no interfaces=іpip-tunne7 networks=192.168.100.24/30 priority=1 type=ptp
add area=backbone cost=40 disabled=no interfaces=іpip-tunne8 networks=192.168.100.28/30 priority=1 type=ptp
```

Рисунок 3 – Конфігурація OSPF підрозділу 1

```
/routing ospf instance
add disabled=no name=default-v2 redistribute=connected,static,vpn,dhcp,modem
/routing ospf area
add area-id=0.0.0.0 comment="backbone" disabled=no instance=default-v2 name=backbone
add area=backbone cost=10 disabled=no interfaces=іpip-tunnel networks=192.168.100.0/30 priority=1 type=ptp
add area=backbone cost=20 disabled=no interfaces=іpip-tunne2 networks=192.168.100.4/30 priority=1 type=ptp
add area=backbone cost=30 disabled=no interfaces=іpip-tunne3 networks=192.168.100.8/30 priority=1 type=ptp
add area=backbone cost=40 disabled=no interfaces=іpip-tunne4 networks=192.168.100.12/30 priority=1 type=ptp
add area=backbone cost=10 disabled=no interfaces=іpip-tunne9 networks=192.168.100.16/30 priority=1 type=ptp
add area=backbone cost=20 disabled=no interfaces=іpip-tunne10 networks=192.168.100.20/30 priority=1 type=ptp
add area=backbone cost=30 disabled=no interfaces=іpip-tunne11 networks=192.168.100.24/30 priority=1 type=ptp
add area=backbone cost=40 disabled=no interfaces=іpip-tunne12 networks=192.168.100.28/30 priority=1 type=ptp
```

Рисунок 4 – Конфігурація OSPF підрозділу 2

```
/routing ospf instance
add disabled=no name=default-v2 redistribute=connected,static,vpn,dhcp,modem
/routing ospf area
add area-id=0.0.0.0 comment="backbone" disabled=no instance=default-v2 name=backbone
/routing ospf interface-template
add area=backbone cost=10 disabled=no interfaces=ipip-tunne5 networks=192.168.100.16/30 priority=1 type=ptp
add area=backbone cost=20 disabled=no interfaces=ipip-tunne6 networks=192.168.100.20/30 priority=1 type=ptp
add area=backbone cost=30 disabled=no interfaces=ipip-tunne7 networks=192.168.100.24/30 priority=1 type=ptp
add area=backbone cost=40 disabled=no interfaces=ipip-tunne8 networks=192.168.100.28/30 priority=1 type=ptp
add area=backbone cost=10 disabled=no interfaces=ipip-tunne9 networks=192.168.100.16/30 priority=1 type=ptp
add area=backbone cost=20 disabled=no interfaces=ipip-tunne10 networks=192.168.100.20/30 priority=1 type=ptp
add area=backbone cost=30 disabled=no interfaces=ipip-tunne11 networks=192.168.100.24/30 priority=1 type=ptp
add area=backbone cost=40 disabled=no interfaces=ipip-tunne12 networks=192.168.100.28/30 priority=1 type=ptp
```

Рисунок 5 – Конфігурація OSPF підрозділу 3

У всіх інших варіантах схем конфігурація виглядатиме аналогічно, відрізняючись кількістю тунелів.

Таким чином, операційна система RouterOS є перспективним та універсальним рішенням, що дозволяє ефективно вирішувати проблеми доступності зв'язку між віддаленими підрозділами різних організацій. За наявності кількох провайдерів у кожному з підрозділів можна протягом декількох секунд відновлювати локальну мережу, використовуючи резервні тунелі. Завдяки тому, що ця операційна система динамічно розвивається, її можливості постійно розширюються та покращуються.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *RouterOS - RouterOS - MikroTik Documentation*. (б. д.). MikroTik Routers and Wireless - Support. <https://help.mikrotik.com/docs/>
2. Мой, J. Т. (1998). *OSPF: anatomy of an Internet routing protocol*. Addison-Wesley Professional.
3. Towidjojo, R. (2023). *Mikrotik Kung Fu: Kitab 4*. Jasakom.

РОЗВИТОК ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНОЇ ПЛАТФОРМИ ДЛЯ ПРОЦЕСІВ УПРАВЛІННЯ ДОКУМЕНТАЦІЄЮ ТРАНСПОРТНИХ МЕРЕЖ

Бабенко В.О., Лантрат О.В.

Харківський національний автомобільно-дорожній університет, Харків

Community cloud - це багатоклієнтський хмарний сервіс, який розділений між кількома підприємствами і управляється та захищається в основному всіма привабливими підприємствами. Він підтримує нішеве співтовариство, яке має спільні цілі (наприклад, мета, потреби в безпеці). Це гібридна форма приватної хмари. Розглянемо деякі з основних сценаріїв моделі Community Cloud, які є вигідними для організацій-учасників.

- Кілька урядових відомств, які здійснюють транзакції один з одним, можуть мати свої системи обробки на спільній інфраструктурі.
- Багатьом компаніям може знадобитися певна система або додаток, розміщені на хмарних сервісах. Хмарний провайдер дозволяє різним користувачам підключатися до одного середовища і логічно сегментувати свої сесії.
- Агентства можуть використовувати цю модель для тестування програм з високими потребами безпеки, а не використовувати Public Cloud.

Позитиви хмарних моделей спільноти включають в себе:

- Покращені послуги
- Безпека та відповідність
- Надійність
- Гнучкість і масштабованість

Проаналізуємо виклики та можливості, розглянуті в реалізації Cloud (табл. 1).

Таблиця 1 – Виклики та можливості, розглянуті в реалізації Cloud

Виклики	Можливості
Організаційна інерція Відсутність досвіду в хмарних обчисленнях Проблеми безпеки Вузькі місця інфраструктури	Фінансова підтримка закупівель обладнання Потреби ІТ-інфраструктури

Саджу Метью провів дослідження під назвою «Впровадження хмарних обчислень в освіті» – революція. Згідно з висновками, багато університетів удосконалили фундаментальні структури ІТ та даних. Завдання можуть бути вирішені за допомогою хмарних обчислень. Ці виклики включали витрати, гнучкість, доступ до інформації та технологій. Використовуючи технологію хмарних обчислень в установах, слід визначити необхідність, а ідентифікація конфіденційності та хмарні сервіси такі:

1) Інфраструктура як послуга (IaaS): може бути використаний для задоволення інфраструктурних потреб студентів, факультетів або дослідників в усьому світі або на місцевому рівні з апаратною конфігурацією для конкретного завдання.

2) Платформа як Послуга (PaaS): деякі провайдери відкривають платформи додатків, щоб дозволити клієнтам створювати свої власні додатки без витрат і складності покупки і управління основними апаратними і програмними рівнями.

3) Програмне забезпечення як послуга (SaaS): постачальник послуг додатків розміщує програму, яка працює і взаємодіє через веб-браузер, розміщений настільний комп'ютер або віддалений клієнт. Це виключає необхідність установки і запуску програми на власному комп'ютері клієнта та спрощення технічного обслуговування та підтримки.

4) Обчислення як послуга (CaaS): провайдери пропонують доступ до необроблених обчислювальних потужностей на віртуальному сервері, таких як Amazons, EC2. На наступному малюнку показаний університет за допомогою сервісів хмарних обчислень.

Системи, що використовують хмарні обчислювальні послуги, мають наступні задачі:

1) Синтезувати особливості управлінської інформаційної системи в транспортній мережі за допомогою технології хмарних обчислень.

2) Синтезувати елементи управлінської інформаційної системи в транспортній мережі за допомогою технології хмарних обчислень.

3) Розробити моделі управлінської інформаційної системи в транспортній мережі за технологією хмарних обчислень.

4) Оцінювати моделі управлінської інформаційної системи в транспортній мережі за технологією хмарних обчислень.

Розвиток інформаційної системи управління хмарними обчисленнями в транспортній мережі може бути розділений на чотири етапи:

1) Поєднуйте особливості інформаційної системи управління хмарними обчисленнями в транспортній мережі з пов'язаними документами.

2) Закріпити розміри інформаційної системи управління хмарними обчисленнями в транспортній мережі.

3) Розробити моделі інформаційної системи управління хмарними обчисленнями в транспортній мережі.

4) Аналізувати моделі інформаційної системи управління хмарними обчисленнями в транспортній мережі.

Література

1. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems, 25:599616, 2019.

2. The Transformation of Education through State Education Clouds, www.ibm.com/ibm/files/N734393J24929X18/EBW03002-USEN-00.pdf

МОДЕЛЮВАННЯ ПРОЦЕСУ ОБСЛУГОВУВАННЯ КЛІЄНТІВ У БАНКУ

Масін І.Е.

Харківський національний автомобільно-дорожній університет, Харків

Успішний економічний розвиток держави неможливий без добре розвинутої банківської системи, на стан якої впливає як внутрішня економічна і політична ситуація, так і зміни світової банківської системи. Ці зміни є наслідком глибинних макроекономічних процесів у світовій економіці (інтеграція, лібералізація, різкий науково-технічний підйом) і вони зачіпають всі банки, незалежно від рівня їх розвитку.

Розвиток дистанційного банківського обслуговування має на меті скорочення в Україні готівкових розрахунків, залучення у банківську систему значного обсягу коштів населення. В останні роки український ринок дистанційного банківського обслуговування відбувається не так стрімко як в розвинутих країнах світу.

Декілька років тому основною послугою, пропонованою банками у сфері дистанційного обслуговування, була послуга «Клієнт-Банкінг». Поступово система «Клієнт- банкінг» стала замінюватися системою «Інтернетбанкінг». В даний час багато банківських установ надають клієнтам ці обидві послуги, інші не використовуються зовсім, або з обмеженим функціоналом

Ефективне управління чергами та мінімізація часу очікування клієнтів дозволяють підвищити рівень задоволеності клієнтів та оптимізувати роботу персоналу. Для цього важливим інструментом є моделювання процесів обслуговування клієнтів.

Метою роботи є створення програми, що відображає процес обслуговування клієнтів у відділенні банку. Він допомагає візуалізувати, аналізувати та оптимізувати процеси обслуговування, визначати проблемні області та знаходити рішення.

Алгоритм роботи

Основні параметри моделі включають:

Прихід клієнтів: відбувається через випадкові інтервали часу із заданими параметрами.

Час обслуговування: випадковий, також із заданими параметрами.

Алгоритм роботи:

Прихід клієнта: Клієнт входить до банку через випадковий проміжок часу.

Вибір обслуговування: Клієнт обирає між касиром та банкоматом.

Касир: Якщо хоча б один касир вільний, клієнт одразу потрапляє до нього.

Черга: Якщо всі касири зайняті, клієнт стає у найкоротшу чергу.

Принцип обслуговування: "Першим прийшов – першим обслужений".
Вихід з банку: Після обслуговування клієнт залишає банк.

Опис програмного функціоналу процесу обслуговування клієнтів у банку

Програма створена у вигляді бота який управляє клієнтами в банку, дозволяючи їм ставати в чергу до касирів або банкоматів. Він асинхронно обробляє ці черги, сповіщаючи користувачів, коли їхнє обслуговування завершено.

Основні компоненти включають завантаження/збереження даних, обробку команд для взаємодії з користувачами та безперервний цикл, що обробляє послуги та оновлює користувачів.

Програма має наступні основні функціональні компоненти:

Генерація клієнтів:

– прихід клієнтів моделюється за допомогою випадкових інтервалів часу з визначеними параметрами.

– вхід клієнта в банк через випадковий проміжок часу. вибір типу обслуговування:

– клієнт обирає між касиром та банкоматом.

– якщо хоча б один касир вільний, клієнт одразу потрапляє до нього. якщо всі касири зайняті, клієнт стає у найкоротшу чергу.

Процес обслуговування:

– принцип обслуговування клієнтів реалізований як "першим прийшов – першим обслужений".

– час обслуговування є випадковим і задається з визначеними параметрами.

Вихід з банку:

– після завершення обслуговування клієнт залишає банк. моделювання черг:

– використовується теорія черг для моделювання процесів обслуговування клієнтів.

– включає визначення кількості кас та банкоматів (в даній моделі 3 каси та 8 банкоматів).

– враховуються випадкові інтервали прибуття клієнтів та випадковий час обслуговування.

Перспективи розвитку

Серед перспектив розвитку системи можна застосовувати наступні технології:

Вдосконалення стратегій управління чергами

Одним з ключових аспектів розвитку моделі є аналіз та впровадження різних стратегій управління чергами. На практиці це може включати:

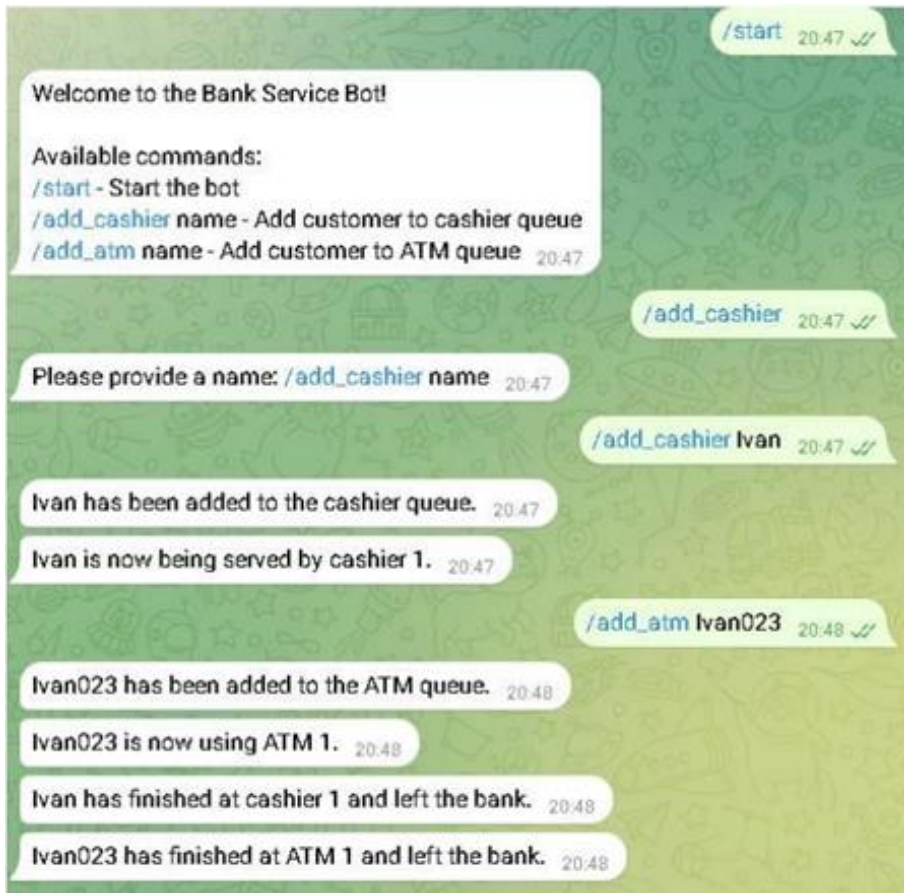


Рисунок 1 – Вигляд головного вікна бота

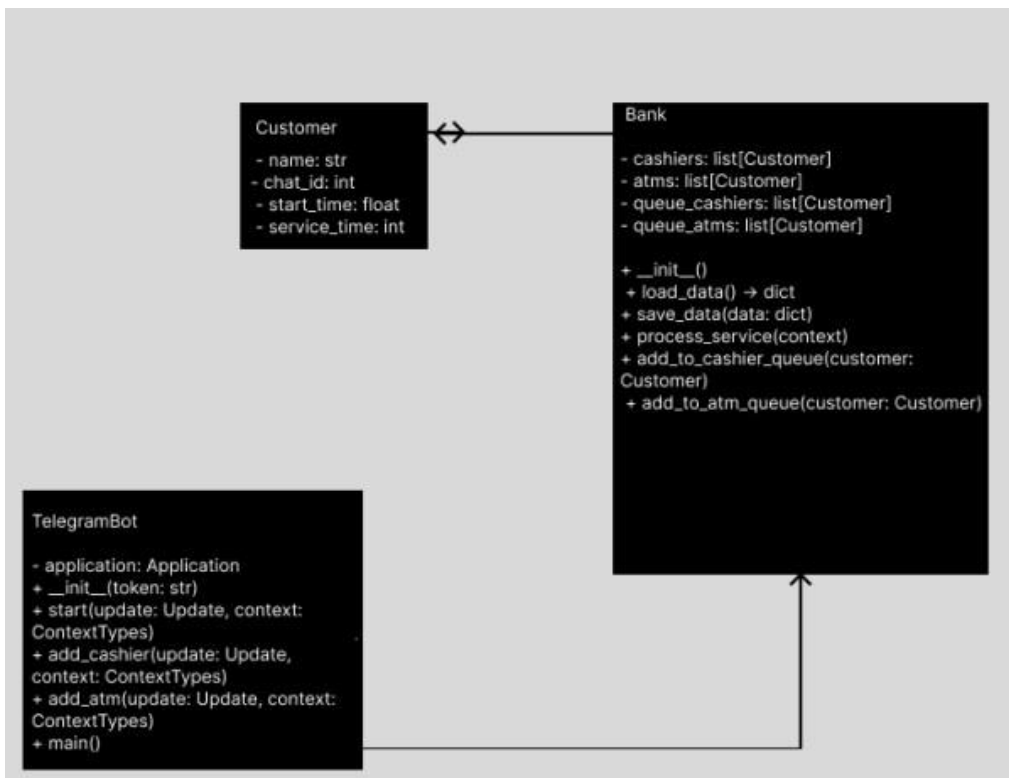


Рисунок 2 – Діаграма класів

Динамічне управління чергами: Впровадження алгоритмів, які в реальному часі змінюють розподіл клієнтів між касами та банкоматами залежно від поточного навантаження.

Пріоритетні черги: Створення окремих черг для клієнтів з різними рівнями пріоритетності, що дозволить швидше обслуговувати VIP- клієнтів або клієнтів з обмеженими можливостями.

Черги з обмеженим часом очікування: Впровадження системи, яка гарантує обслуговування клієнта протягом певного часу, що може зменшити негативний вплив тривалого очікування.

Впровадження додаткових параметрів

Для підвищення точності та адаптивності моделі можна додати нові параметри, які відобразять реалії банківського обслуговування:

Поведінкові параметри клієнтів: Модель може враховувати типові поведінкові патерни клієнтів, такі як частота відвідувань, вибір між касами та банкоматами, рівень задоволеності послугами.

Поточне завантаження: Інтеграція даних про поточне завантаження кас та банкоматів для точнішого прогнозування часу очікування та обслуговування.

Сезонні та денні коливання: Врахування сезонних та денних коливань в потоці клієнтів для оптимального планування ресурсів.

Інтеграція з іншими системами

Розширення моделі може включати інтеграцію з іншими банківськими системами та сервісами:

CRM-системи (Customer Relationship Management): Взаємодія з системами управління відносинами з клієнтами для персоналізації обслуговування та підвищення його якості.

Системи аналітики: Включення аналітичних інструментів для моніторингу ефективності роботи касирів та банкоматів, а також для аналізу поведінки клієнтів.

Онлайн-банкінг та мобільні додатки: Синхронізація з онлайн- банкінгом та мобільними додатками для управління чергами, де клієнти можуть забронювати час для відвідування або отримати інформацію про поточний стан черги.

Висновки

Моделювання процесу обслуговування клієнтів у банку дозволяє краще зрозуміти динаміку черг та ефективно управляти ресурсами.

Такий підхід допомагає банкам забезпечити високу якість обслуговування, що є важливим чинником конкурентоспроможності на ринку фінансових послуг.

Література

1. Бібліотека ім. Л. Каніщенка [Електронний ресурс]. Режим доступу: <http://library.wunu.edu.ua/> Дата доступу: 15.04.2024
2. Про електронний цифровий підпис [Електронний ресурс]. Режим доступу: <https://zakon.rada.gov.ua/laws/show/852-15#Text> <https://bank.gov.ua>. Дата доступу: 15.04.2024

ІНФОРМАЦІЙНА СИСТЕМА ВИБОРУ СТАНЦІЇ ТЕХНІЧНОГО ОБСЛУГОВУВАННЯ

Олена Шапошнікова¹, Дмитро Олійник²

¹Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaep@gmail.com

²Харківський національний автомобільно-дорожній університет, Харків.

В сучасному світі кількість автомобілів сильно зросла та невпинно збільшується. Майже кожна 4 людина в країні має власний автомобіль. Разом з попитом на автомобілі, зросли і запити на технічне обслуговування та ремонт автомобілів. Тому в технічній парадигмі автомобільної індустрії надзвичайно важливою стає роль сервісних та ремонтних центрів, відомих як СТО (станції технічного обслуговування). Оптимізація процесів та підвищення ефективності цих установ неможливі без застосування сучасних технологій у сфері їхньої діяльності. Однією з ключових інновацій у цьому контексті є технології пошуку СТО. Ці технології базуються на використанні сучасних інформаційних та комунікаційних засобів для забезпечення ефективної взаємодії між власниками автомобілів та сервісними центрами.

Інформаційні системи (ІС) для вибору і пошуку СТО допомагають відшукати, переглянути та обрати станції технічного обслуговування, які задовольняють запит користувачів.

На сьогодні існує чи немало систем пошуку, які дозволяють знайти місце знаходження потрібної СТО, переглянути корисну інформацію та відгуки про неї. Деякі з них використовують технології систем онлайн-платформ та мобільних застосунків, інтерактивних систем планування та керування графіками обслуговування, системи діагностики.

Аналізуючи подібні проекти можна з упевненістю зробити висновок, що система пошуку СТО є необхідною для користувачів автомобілів та власників сервісних центрів. Дані системи та вибору допомагають користувачам швидко та ефективно знайти та обрати сервісний центр, якій зможе задовольнити запит клієнта [1-2]. Також ці системи допомагають сервісним центрам покращити ефективність процесів планування графіків робіт відповідно до запитів клієнтів.

Деякі вже існуючі системи мають на меті полегшення та оптимізацію процесу знаходження та вибору сервісного центру для обслуговування автомобілів. Такі системи або сервіси мають наступні функції:

- пошук за місцезнаходженням;
- фільтрація за параметрами;
- перегляд інформації про сто;
- рейтинг та відгуки користувачів.

Також такі системи можуть мати різні додаткові функції, такі як:

- онлайн запис та керування графіками;
- нагадування та сповіщення;

- інтеграція з GPS;
- оцінка якості обслуговування.

Через події в країні велика кількість людей були вимушені змінити місце проживання. Деякі сервіси були вимушені припинити свою діяльність або змінити місце своєї дислокації, а якісь тільки почали своє функціонування. Більшість інформації про потрібні сервіси технічного обслуговування, їх компетенцію та якість обслуговування людина дізнається з так званого «сарафанного радіо». А в деяких критичних ситуаціях власники авто можуть не мати потрібної інформації або знаходитися в незнайомому районі чи області. Саме тому, так важливо мати актуальну інформацію про СТО кожному користувачеві автомобілем і не тільки.

Зважаючи на всю актуальність даної теми, було обрано метою роботи розроблення інформаційної системи (ІС) для пошуку сервісних центрів технічного обслуговування яка має забезпечити зручність та ефективність процесів знаходження та вибору потрібного користувачеві сервісного центру, враховуючі аспекти зручності та доступності актуальної інформації для користувачів різних категорій [3-4].

Для досягнення поставленої мети було вирішено такі задачі. В рамках аналізу системи, було визначено цільову аудиторію ІС та сформульовані вимоги до системи. Для фіксації обсягу проєкту, визначення для кого і навіщо потрібна розробка ІС та що потрібно для цього в контексті вимог користувача, було сформульоване положення про концепцію системи.

Система створюється для власників автомобілів будь-яких марок та моделей, які шукають зручний спосіб знайти найближчі та найбільш підходящі для них СТО для обслуговування та ремонту своїх транспортних засобів. Також система може бути корисна для майстрів та власників сервісних центрів, які можуть реєструвати свої СТО на платформі та залучати нових клієнтів, представляючи свої послуги та переваги.

В процесі розробки ІС було визначено такі вимоги користувача:

- зручний та інтуїтивно зрозумілий інтерфейс;
- достовірна інформація;
- можливість фільтрації та сортування;
- функція онлайн-запису та керування графіком;
- надійність та безпечність;
- мобільна доступність;

Застосування ІС дозволить зробити пошук легким, ефективним та актуальним в зручному середовищі для власників автомобілів та сервісних центрів.

Список використаних джерел

1. Сухий О.Л., Міленін В.М., Тарадайнік В.М. Алгоритми пошуку в інформаційних системах / Сухий О. Л. Алгоритми пошуку в інформаційних системах: методичні рекомендації / О. Л. Сухий, В. М. Міленін, В. М. Тарадайнік. – К., 2015. – 2,0 д.а.
2. Павлиш В., Гліненко Л., Шаховська Н. Основи інформаційних технологій і систем : «Львівська політехніка», – 2018 р.

3. Vesto. [Електронний ресурс] / Режим доступу: <https://vsesto.ua>

4. MotoFocus. [Електронний ресурс] / Режим доступу: <https://motofocus.eu>

РОЗРОБКА WEB-ДОДАТКУ ПО ПРОХОДЖЕННЮ ТЕСТІВ ПРАВИЛ ДОРОЖНЬОГО РУХУ

Полупан Анна Віталіївна, Пронін С.В.

Харківський національний автомобільно-дорожній університет, Харків

Веб-додаток для проходження тесту на знання Правил дорожнього руху (ПДР) розроблено з метою надання користувачам зручного та інтерактивного способу підготовки до іспитів на права водія.

Програма реалізована з використанням технологій Java Script, HTML та CSS, а бізнес-логіка написана за допомогою фреймворку Angular. Додаток взаємодіє з базою даних (DB) для зберігання та управління питаннями тестів та результатами проходження тестів.

Основні функції

Проходження тестів:

Випадкові питання: Питання для тесту вибираються випадково з бази даних, щоб кожен тест був унікальним і користувачі могли краще підготуватися до реальних іспитів.

Типи питань:

Питання із зображеннями: Питання, які потребують аналізу зображень, наприклад, дорожніх знаків або ситуацій на дорозі.

Відповіді із зображеннями: відповіді, в яких потрібно вибрати відповідне зображення згідно з питанням.

Відображення результатів:

Підсумковий результат: Після завершення тесту користувачеві показують результати, включаючи загальну кількість правильних та неправильних відповідей, а також загальний відсоток проходження.

Технологічний стек

Backend:

Java Script: Використовується для створення API, які забезпечують зв'язок між клієнтською частиною програми та базою даних.

Frontend:

Angular: Фреймворк для створення динамічних та інтерактивних інтерфейсів користувача. Angular дозволяє створювати односторінкові програми з високоєфективною роботою на стороні клієнта.

HTML та CSS: Використовуються для розробки адаптивного та інтуїтивно зрозумілого інтерфейсу. HTML забезпечує структуру сторінки, а CSS – стилізацію та візуальну привабливість.

Приклад використання

Проходження тесту:

Крок 1: Користувач відкриває програму та натискає на кнопку Start test.

Крок 2: Програма випадково вибирає питання з бази даних і відображає їх користувачу в інтерактивному інтерфейсі.

Крок 3: Користувач відповідає питанням, вибираючи правильні варіанти відповідей.

Крок 4: Після завершення тесту програма відображає результати, включаючи правильні та неправильні відповіді та відсоток проходження.

Отже: Веб-додаток для проходження тесту ПДР забезпечує зручний та інтуїтивно зрозумілий спосіб перевірки знань користувачів. Використання сучасних технологій, таких як Java Script, MongoDB та Angular, гарантує високу продуктивність, безпеку та розширюваність системи. Додаток допомагає користувачам готуватися до іспитів на знання ПДР, аналізувати свої помилки та покращувати свої знання, надаючи функціональні можливості для проходження тестів та аналізу результатів.

Джерела літератури

1. Приклад розробки додатку на Angular: <https://angular.io/tutorial/first-app>
2. Довідник по CSS: <https://css.in.ua/css/properties>
3. Учебник Express часть 3: Использование базы данных (с помощью Mongoose): https://developer.mozilla.org/ru/docs/Learn/Server-side/Express_Nodejs/mongoose
4. Компоненти інтерфейсу користувача Angular Material: <https://material.angular.io/components/categories>

ПОКРАЩЕННЯ СИСТЕМИ ДЕРЖАВНОГО КЕРУВАННЯ ЗА РАХУНОК ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Соколова А.О., Пронін С.В.

Харківський національний автомобільно-дорожній університет, Харків

Блокчейн – це децентралізована система реєстрації інформації, яка забезпечує прозорість, безпечність та незмінність даних. Застосування цієї технології у сфері державного керування може значно покращити її ефективність, підзвітність та рівень довіри з боку громадян.

Основні тези

1. Прозорість:

Блокчейн може забезпечити прозорість у роботі державних органів, роблячи всі дані про їх діяльність доступними для громадян.

Це допоможе у боротьбі з корупцією та зловживаннями владою.

Приклади:

Публікація даних про бюджет та витрати державних органів.

Відкриття реєстрів власності та земельних ділянок.

Забезпечення доступу до інформації про тендери та держзакупівлі.

2. Безпечність:

Блокчейн – це дуже безпечна система, яка захищає дані від несанкціонованого доступу та фальсифікації.

Це може значно покращити захист персональних даних громадян та запобігти кіберзлочинам.

Приклади:

Зберігання персональних даних громадян у зашифрованому вигляді.

Захист даних про голосування на виборах.

Забезпечення безпеки електронних документів.

3. Ефективність:

Блокчейн може автоматизувати багато рутинних процесів, що значно скоротить час та витрати на їх виконання.

Це допоможе зробити державне керування більш ефективним та економним.

Приклади:

Автоматизація процесу реєстрації власності.

Спрощення процедури отримання дозволів та ліцензій.

Зменшення бюрократії та паперової роботи.

4. Підзвітність:

Блокчейн може забезпечити підзвітність державних органів перед громадянами.

Завдяки цій технології громадяни зможуть бачити, як використовуються їхні податки та інші ресурси.

Приклади:

Відстеження бюджетних витрат та цільового використання коштів.

Моніторинг роботи державних органів та службовців.

Забезпечення зворотного зв'язку від громадян до влади.

5. Довіра:

Застосування блокчейну може значно підвищити рівень довіри громадян до влади.

Це допоможе у побудові більш відкритого та демократичного суспільства.

Приклади:

Збільшення участі громадян у прийнятті рішень.

Зменшення рівня корупції та зловживань владою.

Підвищення рівня задоволеності громадян роботою державних органів.

Платформи для побудови блокчейн-мереж

Існує кілька платформ для побудови блокчейн-мереж, кожна з яких має свої переваги та недоліки. Ось кілька платформ разом з їхніми характеристиками:

Ethereum:

Плюси: Популярна та широко використовується платформа з можливістю виконання смарт-контрактів, що робить її підходящою для різноманітних застосувань.

Мінуси: Проблеми з масштабованістю та високі комісії за транзакції можуть стати проблемою у великомасштабних системах.

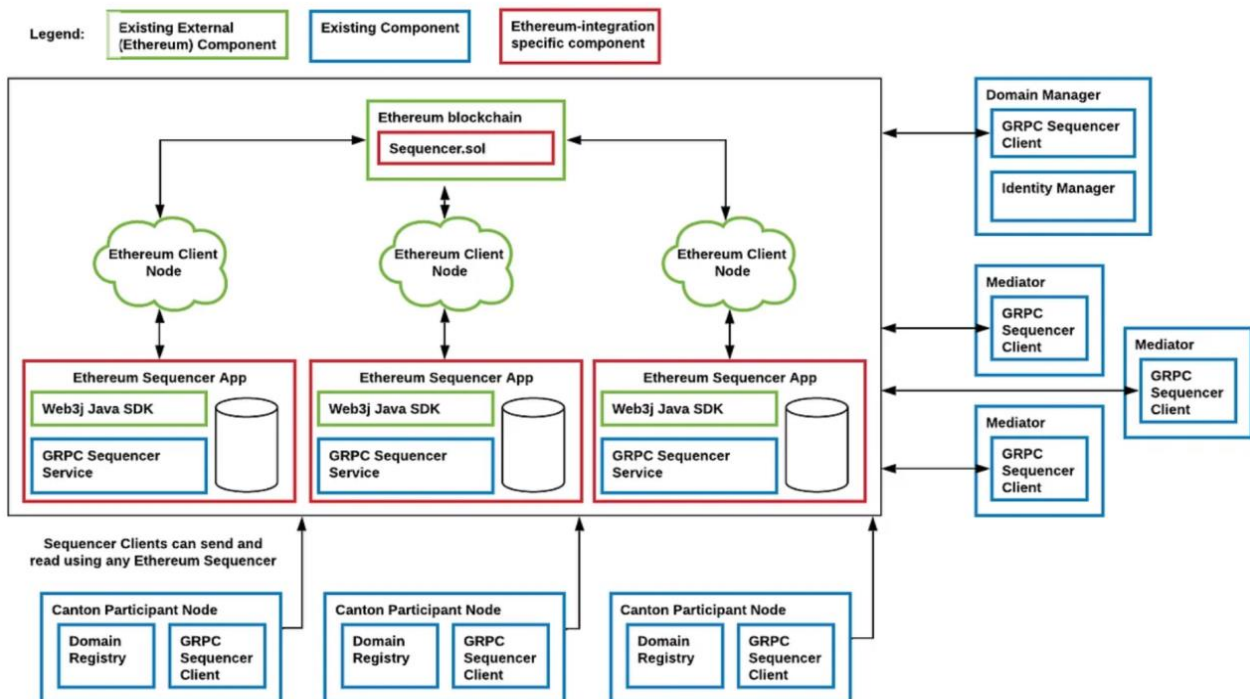


Рисунок 1 – Мережа Ethereum

Hyperledger Fabric:

Плюси: Розроблена спеціально для корпоративних застосувань з підтримкою приватних та публічних мереж. Має високий рівень конфіденційності та доступу до даних.

Мінуси: Складність використання та менша гнучкість порівняно з іншими платформами.

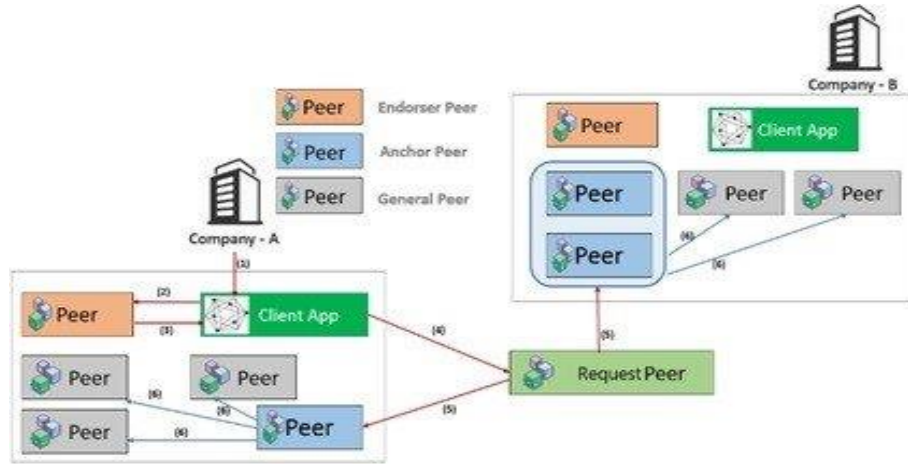


Рисунок 2 – Мережа Hyperledger Fabric

Corda:

Плюси: Створена для фінансових установ з високим рівнем приватності та конфіденційності.

Мінуси: Менша спільнота користувачів порівняно з іншими платформами, обмежена в інших галузях.

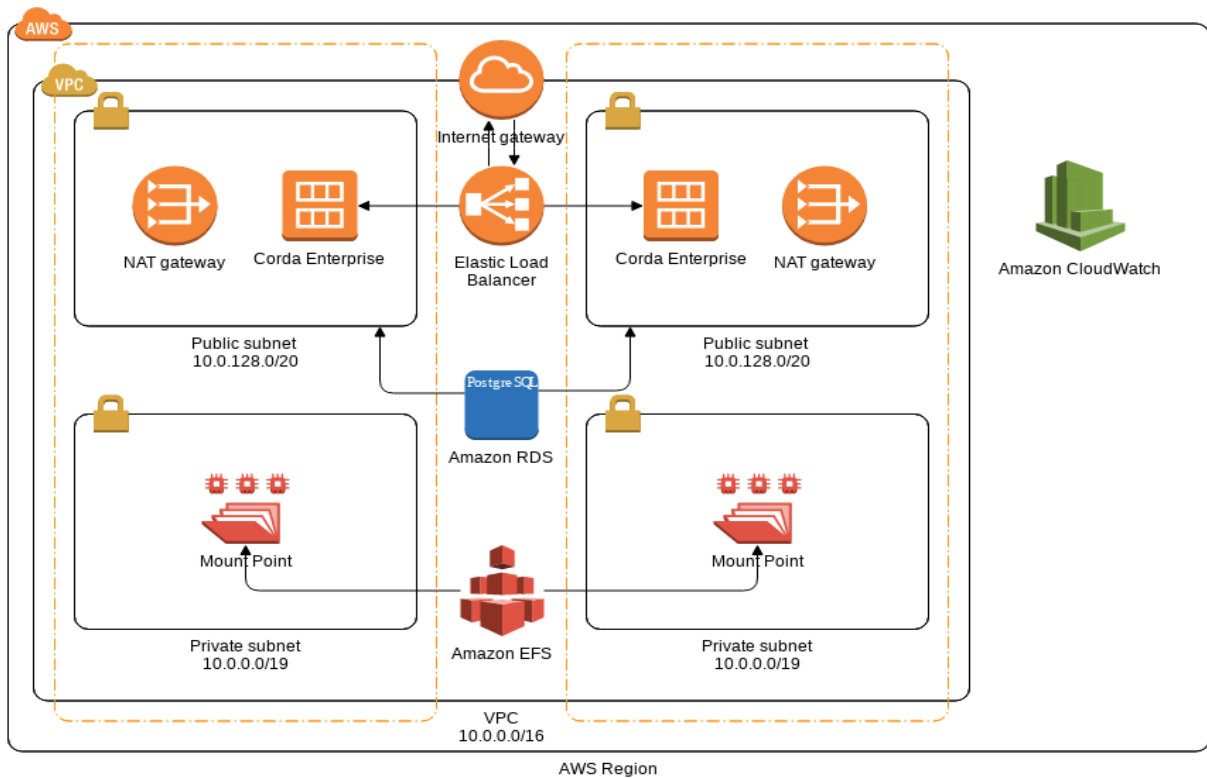


Рисунок 3 – Мережа Corda

Stellar:

Плюси: Орієнтована на міжнародні платежі та фінансові послуги з високою швидкістю та низькими комісіями.

Мінуси: Менша гнучкість та можливості порівняно з Ethereum або Hyperledger.

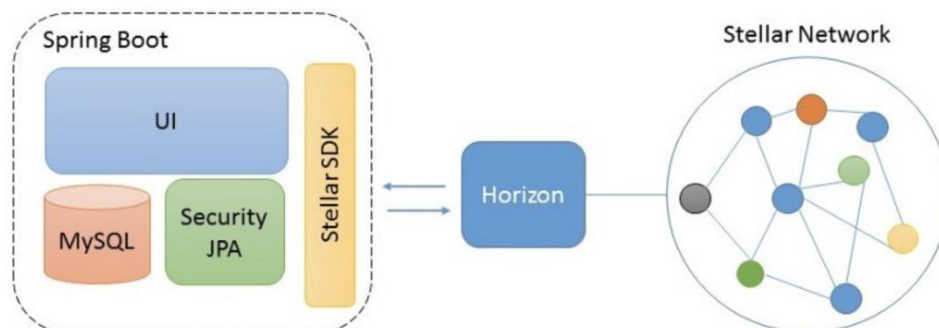


Рисунок 4 – Мережа Stellar

Вибір платформи та архітектури мережі для покращення системи державного керування залежить від конкретних потреб та вимог проекту.

Варіанти будовання мережі блокчейн:

Публічний блокчейн (Public Blockchain)

Відкритий для всіх, де кожен може приєднатися, перевіряти та записувати транзакції.

Приватний блокчейн (Private Blockchain)

Обмежений доступом, зазвичай використовується в корпоративних середовищах.

Консорціумний блокчейн (Consortium Blockchain)

Контрольований групою організацій, які спільно управляють мережею.

Вибір архітектури та платформи для системи державного керування

Вибір архітектури та платформи для системи державного керування через блокчейн залежить від ряду факторів, таких як масштаб проекту, потреби в приватності даних, потенціал для масштабування та технічні ресурси, які Доступні для розробки та підтримки мережі. Однак, ось декілька рекомендацій:

Платформа: З урахуванням комплексності та важливості для державного керування, Hyperledger Fabric може бути добрим вибором. Вона спеціально розроблена для корпоративних застосувань, має високий рівень приватності та конфіденційності даних, а також підтримує масштабованість.

Архітектура: У залежності від конкретних потреб системи державного керування, рекомендовано розглядати модель "Private Permissioned Blockchain" на базі Hyperledger Fabric. Це дозволить забезпечити контроль над доступом до мережі та конфіденційністю даних, що особливо важливо для державних установ.

Приватність даних та безпека: При виборі платформи та архітектури необхідно врахувати високий рівень захисту даних, оскільки це державна інформація. Hyperledger Fabric має вбудовані механізми конфіденційності та безпеки, які дозволяють контролювати доступ до даних та забезпечувати їхню цілісність.

Масштабованість: Hyperledger Fabric також має потенціал для масштабування, що є важливим для систем державного керування, які можуть включати велику кількість учасників та обсягів даних.

Враховуючи ці фактори, Hyperledger Fabric може бути відмінним вибором для системи державного керування через блокчейн, забезпечуючи високий рівень приватності, безпеки та масштабованості. Однак перед остаточним рішенням рекомендується провести докладний аналіз вимог та ресурсів проекту для забезпечення найкращого вибору.

Література

1. Блокчейн [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/ Блокчейн](https://uk.wikipedia.org/wiki/Блокчейн). Дата доступу: 15.04.2024
2. Ethereum [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/ Ethereum](https://uk.wikipedia.org/wiki/Ethereum). Дата доступу: 15.04.2024
3. Hyperledger Fabric [Електронний ресурс]. Режим доступу: <https://www.hyperledger.org/projects/fabric>. Дата доступу: 15.04.2024
4. Corda [Електронний ресурс]. Режим доступу: <https://corda.net>. Дата доступу: 15.04.2024
5. Stellar [Електронний ресурс]. Режим доступу: <https://uk.wikipedia.org/wiki/Stellar>. Дата доступу: 15.04.2024

ВИКОРИСТАННЯ ТЕХНОЛОГІЇ REACT.JS ДЛЯ СТВОРЕННЯ СУЧАСНИХ ВЕБ ЗАСТОСУНКІВ

Олена Шапошнікова¹, Герман Степанов²

¹Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaer@gmail.com

²Харківський національний автомобільно-дорожній університет, Харків

У світі сучасного програмування існує безліч інструментів та технологій, які допомагають розробникам створювати потужні та ефективні програмні рішення. Одним із найбільш популярних інструментів для розробки веб-додатків є бібліотека React.js.

React.js – це JavaScript-бібліотека, розроблена компанією Facebook, яка дозволяє побудувати інтерактивні та ефективні інтерфейси користувача. Однією з ключових особливостей React є його компонентний підхід до розробки, що

дозволяє розділити користувацький інтерфейс на невеликі, повторно використовувані частини, які називаються компонентами.

Переваги React

1. Віртуальний DOM - Віртуальний DOM (VDOM) (рис. 1) – це концепція програмування, в якій ідеальне чи «віртуальне» представлення інтерфейсу користувача зберігається в пам'яті і синхронізується зі «справжнім» DOM за допомогою бібліотеки, такої як ReactDOM. Цей процес називається узгодженням [1, 2]. React використовує віртуальний DOM, що дозволяє зменшити кількість операцій зміни реального DOM і підвищує продуктивність додатку.

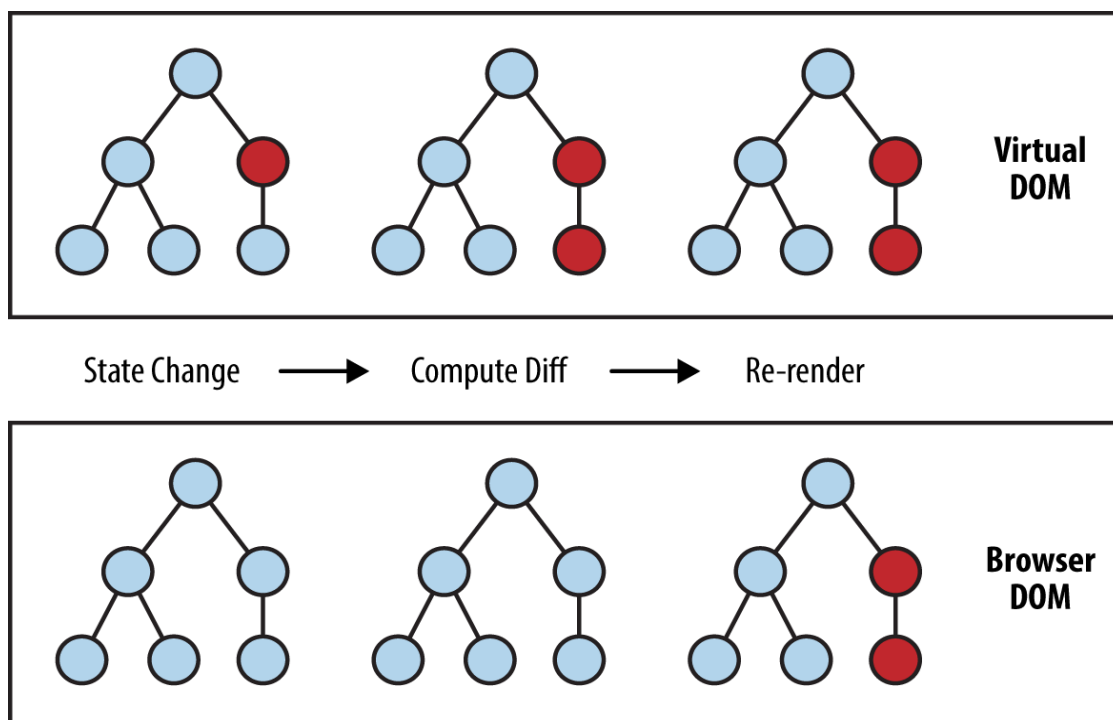


Рисунок 1 – Віртуальний DOM (VDOM)

2. Компонентний підхід – React пропонує чистий та організований спосіб створення веб-інтерфейсів за допомогою компонентів, що полегшує розробку та підтримку додатків. Компоненти дозволяють розділити інтерфейс користувача на незалежні частини, придатні до повторного використання, і сприймати їх як такі, що функціонують окремо один від одного [1].

3. Односторонній потік даних – React пропонує підхід з використанням одностороннього потоку даних, що полегшує розуміння та управління станом додатку. Це концепція, в якій дані передаються вниз по ієрархії компонентів, забезпечуючи їхню чітку та передбачувану взаємодію. Це означає, що дані передаються вниз від батьківських компонентів до дочірніх через властивості (props) [1].

4. Велика спільнота та екосистема – React має велику та активну спільноту розробників, а також широкую екосистему інструментів та бібліотек, що доз-

воляє легко знаходити рішення для будь-яких задач. Є можливість створення та публікацій своїх бібліотек.

Було зроблено порівняльний аналіз React з двома іншими популярними фреймворками, такими як Angular і Vue.js.

React і Angular – це два різних підходи до розробки веб-застосунків.

Angular – це повноцінний фреймворк, який надає багато готових рішень і стандартизує розробку, в той час як React – це бібліотека, яка надає більшу гнучкість і вільний вибір інструментів. Angular має вбудований механізм для керування станом додатку (наприклад, через сервіси та ін'єкцію залежностей), в той час як у React ви можете вибрати будь-яку бібліотеку для керування станом (наприклад, RTK або MobX).

React і Vue.js – подібні за підходом до компонентної розробки та використання віртуального DOM, але вони мають свої відмінності. Vue.js вважається більш простим для початківців, оскільки має менш круту криву навчання, а React вважається більш потужним та масштабованим. React має більшу спільноту та більше інструментів, але Vue.js дозволяє швидше розпочати роботу та зосередитися на розробці. Обидва фреймворки мають свої плюси та мінуси, і вибір між ними залежить від потреб та вподобань розробника.

У підсумку, React – це потужна та водночас гнучка технологія для створення інтерактивних веб-застосунків, яка має свої унікальні переваги порівняно з іншими фреймворками.

Список використаних джерел

1. Документація React [Електронний ресурс] – Режим доступу: <https://react.dev/learn>
2. Стаття про VDOM [Електронний ресурс] – Режим доступу: <https://tonynguyenit.medium.com/how-react-virtual-dom-decide-to-update-browser-dom-91f170718733>
3. Рисунок VDOM [Електронний ресурс] – Режим доступу: <https://alquimiacoffee.medium.com/reacts-virtual-dom-your-new-best-friend-60cf067d8bfd>

АНАЛІЗ НАЙПОПУЛЯРНІШИХ CMS ДЛЯ СТВОРЕННЯ ІНТЕРНЕТ-МАГАЗИНІВ

Стефанців А.В., Лебединський А.В.

Харківський національний автомобільно-дорожній університет, Харків

З кожним роком інтернет стає все більш важливим майданчиком для комерції. З цього приводу, створення інтернет-магазинів стає необхідністю для багатьох бізнесів.

Проте вибір платформи для створення такого магазину може бути складним завданням через велику кількість пропозицій на ринку. У даній роботі ми проведемо аналіз найпопулярніших CMS для створення інтернет-магазинів з метою визначити їх переваги, недоліки та відповідність конкретним потребам бізнесу.

Метою даної роботи є проведення докладного аналізу найпопулярніших CMS для створення інтернет-магазинів з метою визначення їхньої ефективності та придатності для різних видів бізнесу.

Огляд існуючих точок зору. В науковій сфері існує безліч публікацій та досліджень, присвячених аналізу різних CMS для створення інтернет-магазинів. Багато авторів відзначають важливість правильного вибору платформи для успішного функціонування бізнесу в онлайн-середовищі. Проте, не завжди вдається знайти об'єктивний аналіз, який би враховував різні потреби та особливості різних бізнесів.

Передбачувані дослідження та основні результати. Ми передбачаємо, що після проведення аналізу ми зможемо визначити найкращі CMS для різних типів бізнесу.

Основними результатами будуть визначення переваг і недоліків кожної з платформ, їхньої відповідності конкретним вимогам бізнесу та рекомендації щодо вибору найбільш підходящої CMS.

Власні думки та інструменти дослідження. У цій роботі ми використували власний досвід роботи з різними CMS, а також аналіз відгуків користувачів та експертів. Ми також звернулися до статистичних даних та досліджень з відкритих джерел для отримання об'єктивної інформації.

Переваги *Magento*:

- володіє потужними функціями управління та налаштуванням продуктів, замовлень, оплати та доставки [1].
- велика спільнота користувачів та розробників, що забезпечує швидке вирішення проблем та великий обсяг додаткових модулів.

Недоліки *Magento*:

- високі вимоги до хостингу та ресурсоємність, що може збільшувати витрати на інфраструктуру;

- складна установка та налаштування, що вимагає досвіду у роботі з веб-технологіями.

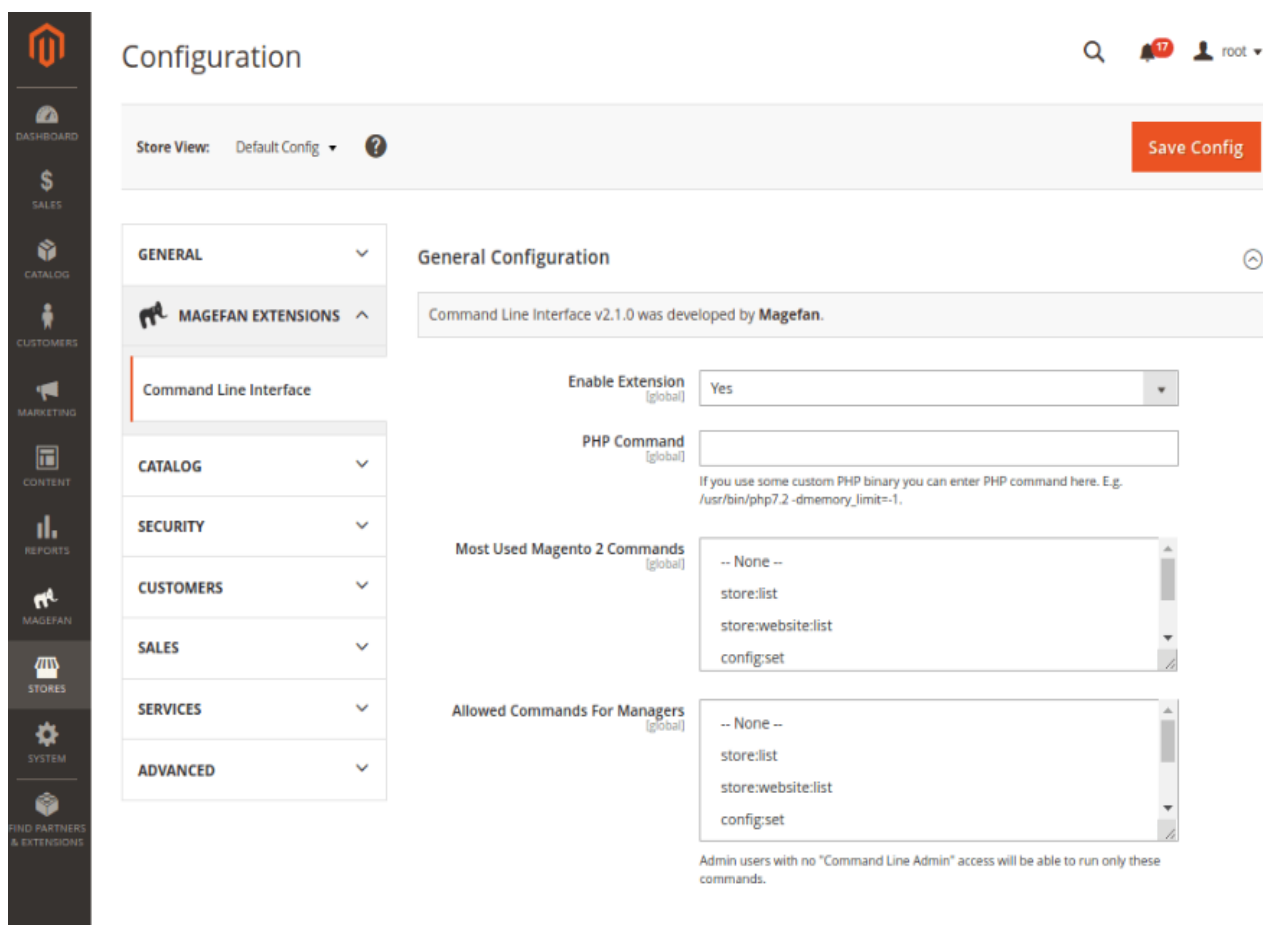


Рисунок 1 – Інтерфейс платформи Magento

Переваги WooCommerce:

- інтеграція з WordPress, що робить його доступним для користувачів, які вже знайомі з цією платформою;
- безкоштовний основний функціонал та можливість розширення за допомогою додаткових плагінів [2].

Недоліки: WooCommerce:

- може стати повільним зі збільшенням кількості товарів та відвідувачів;
- залежність від інших плагінів для розширення функціоналу може збільшувати ризик конфліктів.

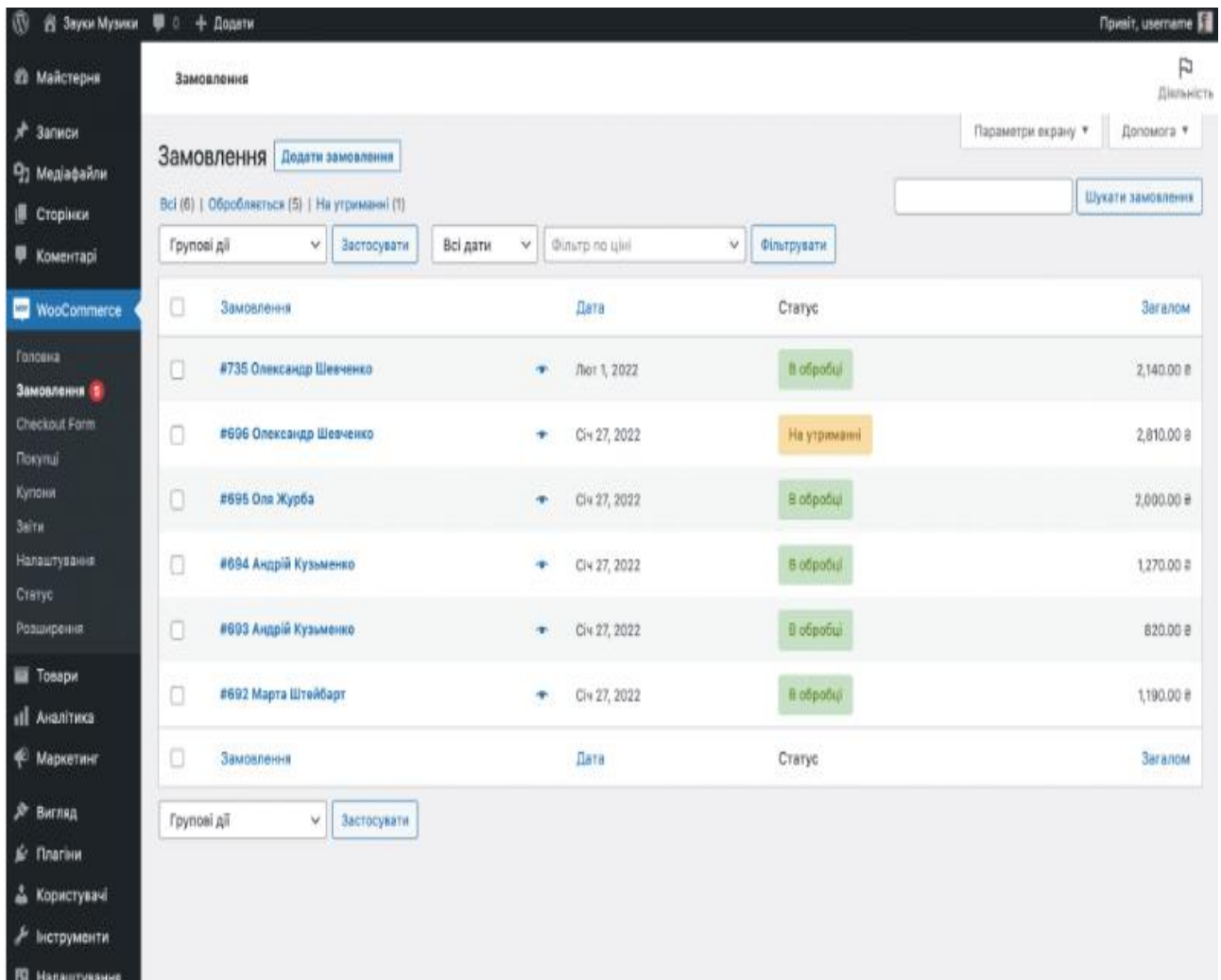


Рисунок 2 – Інтерфейс платформи WooCommerce

Переваги Shopify:

- простий у використанні та налаштуванні, немає потреби у технічних навичках для створення магазину [3];
- широкий вибір готових шаблонів та додатків для розширення функціоналу.

Недоліки Shopify:

- залежність від платних планів та комісій за транзакції, що може збільшувати витрати на довгострокову перспективу;
- обмежена можливість налаштування, особливо для користувачів, які шукають велику гнучкість у кастомізації.

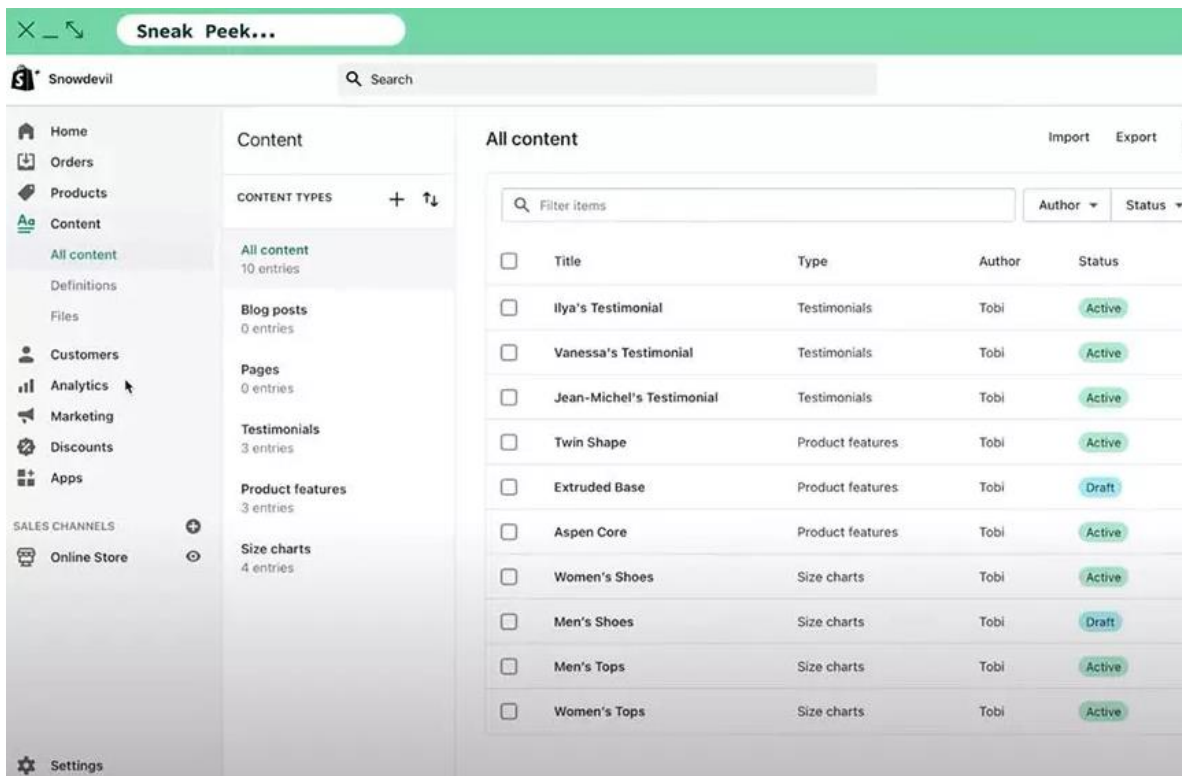


Рисунок 3 – Інтерфейс платформи Shopify

Переваги OpenCart:

- легкий у встановленні та використанні, підходить як початківцям, так і досвідченим користувачам;
- велика спільнота користувачів та багато додаткових модулів для розширення функціоналу [4, 5].

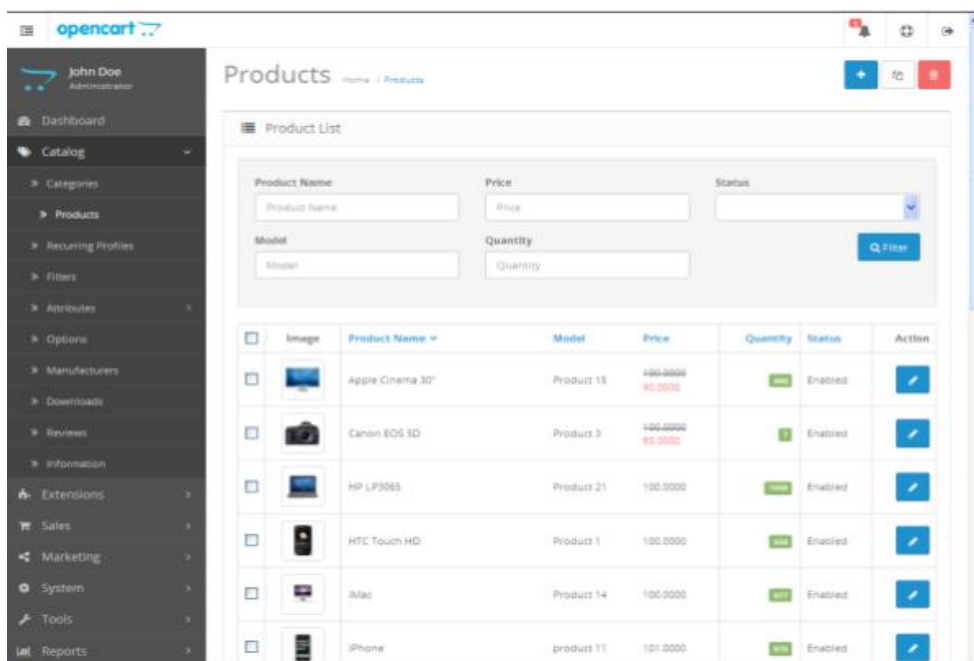


Рисунок 4 – Інтерфейс платформи OpenCart

Недоліки OpenCart:

- не така велика кількість готових шаблонів порівняно з іншими платформами, що може обмежувати дизайн магазину;
- обмежені можливості з підтримки та оновлення програмного забезпечення порівняно з більш великими платформами.

Висновки. На основі проведеного аналізу можна зробити висновок, що вибір CMS для створення інтернет-магазину є важливим кроком для успішного онлайн-бізнесу. Кожна з розглянутих платформ має свої переваги та недоліки, і важливо враховувати їхню відповідність конкретним потребам та можливостям бізнесу.

Список використаних джерел

1. Croll, A., & Shipley, N. "The Definitive Guide to Magento." *Apress*. ISBN: 978-1-4842-4141-5. (date of access 12.04.2024).
2. Williams, L. "WooCommerce 4.0 Cookbook: Transform your WordPress website into a fully-featured e-commerce store with WooCommerce." *Packt Publishing*. ISBN: 978-1-78953-733-1. (date of access 12.04.2024).
3. Pinkerton, L. "Shopify for Dummies." *For Dummies*. ISBN: 978-1-119-51839-3 (date of access 12.04.2024).
4. Lasko, M. "OpenCart 3 Beginners Guide 2018: How to Build Your Own E-commerce Website Step by Step." *Independently published*. ISBN: 978-1980202740 (date of access 12.04.2024).
5. Rayport, J., & Jaworski, B. "Introduction to eCommerce." *McGraw-Hill Education*. ISBN: 978-1-259-09542-8 (date of access 12.04.2024)

УДК 004.942

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОНОМНОЇ НАВІГАЦІЇ НА МІСЦЕВОСТІ

*Хоменко Ю.С., аспірант кафедри метрології та безпеки життєдіяльності,
Харківський національний автомобільно-дорожній університет*

Анотація: В теперішній час є актуальною проблематика навігації автономних мобільних роботів на місцевості за відсутності зв'язку GPS, оскільки остання є дуже актуальною в цей час в умовах війни, а також техногенних і природних катаклізмів. В цей час навігація здійснюється шляхом роботи топологічних зв'язків з координатами об'єктів, котрі можна вважати орієнтирами. Виявлення останніх активними випромінюючими засобами дуже часто неможливо, що пов'язано з впливом фону різних місцевостей. Тому для виконання різних завдань на незнайомій місцевості при відсутності зв'язку GPS можуть використовуватися АМР. АМР повинні виявляти а також розпізнавати різні

об'єкти, які можна віднести до орієнтиру. Декілька орієнтирів є базою для створення системи координат для навігації АМР. Виявлення навігаційних орієнтирів на місцевості доцільно проводити пасивними системами (без випромінювання), зокрема, відеокамерами.

Ключові слова: автономний мобільний робот; орієнтир; методи автономної навігації.

АМР повинні бути здатні самостійно виявити для себе об'єкти, які можна віднести до наземних орієнтирів. Для АМР повинен зробити локальну систему координат, де він може здійснювати навігацію. Створені раніше цифрові карти місцевості у разі несприятливих умов часто стають непридатними. Створення карт потребує багато часу, а іноді і є неможливим в різних умовах. Виявлення орієнтирів АМР в умовах впливу організованих та ненавмисних активних завад, а також пасивних завад, що обумовлені відбиттям електромагнітних хвиль від різних об'єктів, є ускладненим. Звідси, перевагу отримують невипромінюючі бортові системи, тобто, відеокамери.

В доповіді аналізуються методи співставлення зображень на основі яких розробляється інформаційна технологія навігації роботів, а також похибки в роботі, які в наслідок цього виникають та способи їх мінімізації.

Проведено аналіз методів для навігації АМР на основі зображень місцевості, що дає можливість встановити розподіл рішень щодо віднесення зображень до вибраного типу наземних орієнтирів.

Література:

1. Poliarus O. V., Poliakov Ye. O. Detection of Landmarks by Mobile Autonomous Robots Based on Estimating the Color Parameters of the Surrounding Area. - In the book: Oleg Sergiyenko, Wendy Flores-Fuentes, Julio Cesar Rodríguez-Quñonez, (eds.). Examining Optoelectronics in Machine Vision and Applications in Industry 4.0. - IGI Global, 2021, pp. 224-257.
2. Poliarus O. V., Poliakov Ye. O., Lindner L. Determination of landmarks by mobile robot's vision system based on detecting abrupt changes of echo signals parameters. - The 44th Annual Conference of the IEEE Industrial Electronics Society. – Washington D. C., USA, October 21-23, 2018, pp. 3165-3170.
3. T. Fukuda, S. Ito, F. Arai, Y. Yokoyama, Y. Abe, K. Tanaka, Y. Tanaka. Navigation system based on ceiling landmark recognition for autonomous mobile robot-landmark detection based on fuzzy template matching (FTM). - Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots
4. G. Novak; A. Bais; S. Mahlknecht. Simple stereo vision system for real-time object recognition for an autonomous mobile robot. - Second IEEE International Conference on Computational Cybernetics, 2004.

УДОСКОНАЛЕННЯ СЕРВІСІВ У СФЕРІ ОРЕНДИ ТА КУПІВЛІ ЖИТЛА ЗА ДОПОМОГОЮ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Шешунов М.О., Пронін С.В.

Харківський національний автомобільно-дорожній університет, Харків

Вступ

В умовах стрімкого розвитку інформаційних технологій, сфера нерухомості переживає значні зміни. Сучасні веб-платформи, такі як Airbnb, Booking.com революціонізували підходи до оренди і купівлі житла, зробивши ці процеси більш доступними, зручними і ефективними.

Застосування новітніх ІТ-рішень дозволяє не тільки оптимізувати вже існуючі процеси, але й вносить суттєві інновації, які можуть трансформувати весь ринок нерухомості.

Особливо велике значення має використання великих даних та аналітичних інструментів, які дозволяють користувачам отримувати рекомендації, засновані на детальному аналізі поведінки споживачів і ринкових тенденцій. Це змінює підходи до маркетингу і клієнтської взаємодії, роблячи їх більш персоналізованими та цілеспрямованими.

Огляд існуючих рішень

На ринку існує чимало платформ, які забезпечують послуги бронювання житла. Платформи як Airbnb, Booking.com, Vrbo та інші використовують складні алгоритми для відображення пропозицій, що відповідають запитам користувачів. Проте, деякі системи мають недоліки, зокрема, недостатньо гнучкі можливості пошуку, відсутність додаткових опцій для вибору апартаментів за специфічними критеріями та обмежений функціонал зворотного зв'язку.

Одним з основних недоліків існуючих платформ є також відсутність глибокої інтеграції з соціальними мережами та іншими сервісами, що могло б значно покращити користувацький досвід, забезпечивши більш деталізований обмін думками та відгуками.

Технологічні інновації

Новітні ІТ-рішення дозволяють вдосконалювати сервіси за рахунок використання машинного навчання для аналізу поведінки користувачів та автоматизації основних процесів бронювання і оплати. Інтеграція штучного інтелекту може допомогти у визначенні персоналізованих пропозицій, що значно підвищить задоволеність користувачів. Крім того, застосування блокчейн-технологій може забезпечити високий рівень безпеки транзакцій і даних користувачів.

Додатково, розробка мобільного застосунку з інтуїтивно зрозумілим інтерфейсом та вбудованими функціями геолокації та віртуальної реальності

може стати значним кроком у покращенні взаємодії з платформою. Це дозволить користувачам "переглядати" житло на відстані, не виходячи з дому.

Онлайн-платіжні системи

Одним із ключових аспектів вдосконалення системи бронювання житла є інтеграція онлайн-платіжних систем, що забезпечують безпеку, швидкість та зручність транзакцій. Використання таких платформ як Stripe або PayPal дозволяє користувачам здійснювати миттєві платежі без зайвих турбот про безпеку фінансових даних завдяки впровадженню передових технологій шифрування та захисту даних. Також ці системи можуть інтегруватися з мобільними додатками, що робить процес оплати ще зручнішим і доступнішим з будь-якої точки світу.

Блокчейн для забезпечення безпеки транзакцій

Застосування блокчейн-технологій може гарантувати високий рівень безпеки транзакцій і даних користувачів, що є критично важливим для фінансових операцій в сфері нерухомості. Основним прикладом зараз це є оплата криптовалютою.

Розробка системи

Система буде розроблена на базі фреймворку Laravel, що дозволяє забезпечити гнучкість, безпеку і швидкість розробки. Laravel має велику кількість вбудованих функцій, що сприяють швидкому розвитку проєктів: від маршрутизації до роботи з базами даних і кешування.

Інтеграція з соціальними мережами

Інтеграція з соціальними мережами може значно покращити взаємодію користувачів з системою, дозволяючи легко ділитися відгуками та рекомендаціями в їхніх соціальних каналах. Це не тільки сприяє прозорості і довірі, але й стимулює маркетинг з вуст в уста, що є важливим фактором у просуванні сервісів. Впровадження функцій, як-от входу через соціальні мережі, може спростити процес реєстрації та входу в систему, знижуючи бар'єри для нових користувачів.

Використання великих даних для аналізу ринку

Використання технологій великих даних може надати компаніям, що працюють у сфері оренди та купівлі житла, глибші інсайти щодо поведінки користувачів, попиту на певні типи житла, а також трендів ринку. Аналітика великих даних дозволяє оптимізувати ціноутворення, ідентифікувати ключові ринки для розширення та вдосконалювати маркетингові стратегії. Це, в свою чергу, сприяє кращому розумінню потреб користувачів і швидкому реагуванню на зміни в попиті.

Впровадження IoT для покращення управління властивостями

Технології Інтернету речей (IoT) можуть бути інтегровані в системи управління властивостями для автоматизації багатьох процесів, таких як управління енергоспоживанням, безпека житла, та дистанційне керування властивостями. Наприклад, інтелектуальні замки можуть дозволити власникам нерухомості контролювати доступ до апартаментів дистанційно, а датчики руху можуть підвищити безпеку житла. Це не тільки підвищує ефективність управління властивостями, але й забезпечує додатковий комфорт та безпеку для орендарів і власників.

Розвиток мобільних технологій у сфері нерухомості

Розвиток мобільних додатків для бронювання та управління нерухомістю є критичним аспектом удосконалення сервісів. Мобільні додатки можуть надавати користувачам миттєвий доступ до баз даних об'єктів нерухомості, можливість бронювання в реальному часі, а також інструменти для віртуальних турів по апартаментах. Інтеграція з доповненою реальністю може дозволити покупцям або орендарям "відвідувати" житло без фізичного переміщення, що є особливо корисним під час обмежень, пов'язаних з пандемією або для міжнародних інвесторів.

Тестування та впровадження

Тестування системи включатиме юніт-тестування, інтеграційне тестування, та приймальне тестування з реальними користувачами для забезпечення надійності і відповідності функціональних вимог.

На цьому етапі кожен компонент системи буде перевірений окремо для виявлення помилок на ранніх стадіях розробки. Використання фреймворку Laravel дозволяє використовувати вбудовані інструменти, такі як PHPUnit, для написання та виконання юніт-тестів, що забезпечує швидке і ефективне тестування моделей, контролерів та інших компонентів системи.

Після тестування система буде готова до впровадження і масштабування.

Висновки та перспективи розвитку

Інформаційні технології надають потужні інструменти для трансформації традиційних підходів у сфері оренди та купівлі житла. Використання ІТ не тільки сприяє оптимізації існуючих процесів, але й відкриває нові можливості для інновацій, підвищення безпеки, зручності та задоволеності користувачів. Подальше впровадження технологічних нововведень зможе значно підвищити ефективність та якість сервісів, зробивши їх більш конкурентоспроможними і відповідними до потреб сучасних користувачів.

Літературні джерела

1. Ковалюк Д.О., «Проектування інформаційних систем. Частина 2. Web-програмування», КПІ ім. Ігоря Сікорського, 2022

2. Airbnb Engineering Blog – Режим доступу <https://medium.com/airbnb-engineering>
3. Офіційна документація PHP – Режим доступу <http://php.net/manual/en/>
4. Офіційна документація MySQL – Режим доступу <https://dev.mysql.com/doc>
5. Офіційна документація Laravel 11 – Режим доступу <https://laravel.com/docs/11.x/readme>
6. PHP Tutorial W3 – Режим доступу <https://www.w3schools.com/php/default.asp>
7. MySQL Tutorial – Режим доступу <https://www.w3schools.com/sql/default.asp>
8. Laravel News – Режим доступу <https://laravel-news.com/>
9. GitHub – Режим доступу <https://github.com/>

УДК 004.773

ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ СИСТЕМ ПОШУКУ З ВИКОРИСТАННЯМ ХМАРНИХ ТЕХНОЛОГІЙ

Бабенко В.О., Тимченко С.С.

Харківський національний автомобільно-дорожній університет, Харків

Хмарні обчислення набирають популярність завдяки своїй нижчій вартості, високій надійності і величезній доступності. Це динамічно масштабований спосіб доступу до зовнішніх обчислювальних ресурсів як сервісу, що надається Інтернетом, і користувачеві не потрібні ніякі спеціальні знання про інфраструктуру «хмари» або навички управління «хмарної» технології. Теоретично, технологія хмарних обчислень надає загальну інфраструктуру як сервіс – IaaS (Інфраструктура як послуга), Платформа як послуга – PaaS (Платформа як послуга), або програмне забезпечення як послуга – SaaS (Програмне забезпечення як послуга).

Однією з переваг хмарних технологій є можливість масового зберігання даних, Виявлення даних в хмарі стає новою місією для технологій пошукових систем в цьому «хмарному віці», традиційна пошукова система буде стикатися з великим викликом в режимі реального часу пошуку, швидкість відгуку і зберігання масових сторінок, Однак, пошукова система, розгорнута в хмарі, може вирішити ці проблеми пошукова система MapReduce через свою ефективну розподілену обчислювальну базу і розподілену файлову систему HDFS з великим обсягом пам'яті даних. Створення пошукової системи на платформі Hadoop може вирішити проблеми обробки масових даних і зберігання масових даних. І пошукова система буде значно покращена в режимі реального часу пошуку і швидкості відгуку. Хмара може забезпечити обчислювальну потужність. Hadoop MapReduce забезпечує високу масштабованість і стійкість у великому кластері. При розробці пошукової системи в Cloud-середовищі Apache Foundation випустив apache Solr, Solr є відкритою платформою пошуку

й індексації, побудованою на проєкті Lucene. Він полегшує повнотекстовий пошук, виділення збігів та фасетний пошук тексту, багатих документів та геопросторових даних.

Хмара Солера управляється моделлю програмування під назвою Zookeeper, яка є розподіленою службою з відкритим кодом для розподіленої програми. Він являє собою простий набір примітивів, які розподілені додатки можуть побудувати на впровадження більш високого рівня послуг для синхронізації, обслуговування конфігурації, і груп і іменування.

Масштабована система пошуку за допомогою Solr у Cluster дозволяє легко скоротити багато даних, а потім швидко обслуговувати результати за допомогою швидкого, гнучкого пошуку та запиту.

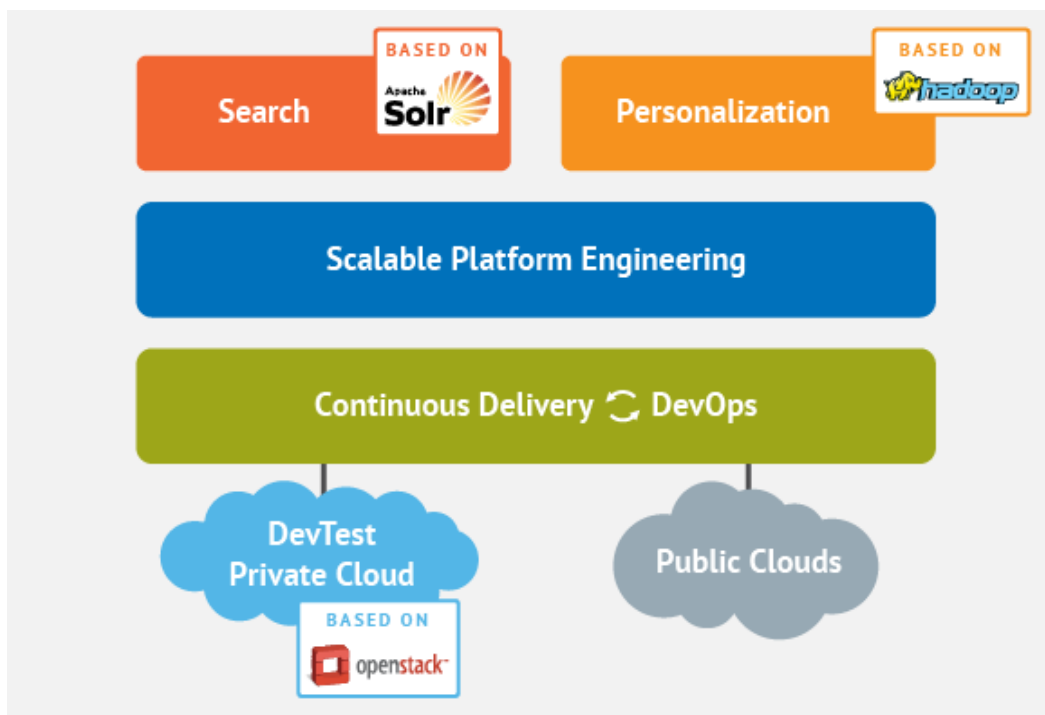


Рисунок 1 – Платформи хмарних технологій

Хмарні технології є виразом, що використовується для опису різних обчислювальних концепцій, які включають в себе велику кількість комп'ютерів, підключених через мережу зв'язку в режимі реального часу, таких як Інтернет. В науці, хмарні обчислення є синонімом розподілених обчислень по мережі, і означає можливість запуску програми або програми на багатьох підключених комп'ютерах одночасно. Фраза також частіше стосується мережевих послуг, які, як видається, надаються реальним серверним обладнанням, і насправді обслуговуються віртуальним апаратним забезпеченням, змодельованим програмним забезпеченням, запущеним на одній або декількох реальних машинах. Такі віртуальні сервери фізично не існують і тому можуть бути переміщені і збільшені (або зменшені) на льоту, не впливаючи на кінцевого користувача - можливо, швидше, як хмара.

Всі компанії хочуть мати можливість заощадити на експлуатаційних витратах. Використання хмарних технологій може допомогти, приймаючи величезні капіталовкладення з рівняння. З темпом технологічних змін це дуже вигідно для невеликих компаній, щоб використовувати серверну ферму, щоб піклуватися про потреби обробки замість того, щоб намагатися зробити все, що в будинку. Багато компаній розглядають це як головну мету при розгляді питання про використання хмарних обчислень; однак є й інші мотивації.

Література:

1. Foster, The grid: Computing without bounds, Scientific American, vol. 288, No. 4, (April 2023), pp. 7885.
2. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems, 25:599616, 2019.
3. The Transformation of Education through State Education Clouds, www.ibm.com/ibm/files/N734393J24929X18/EBW03002-USEN-00.pdf

ОГЛЯД ОСНОВНИХ ЕТАПІВ ПРОЕКТУВАННЯ САЙТУ

*Мацій Ольга Борисівна, доцент кафедри комп'ютерних систем
Харківського національного автомобільно-дорожнього університету*

*Фентисов Богдан, студент механічного факультету
Харківського національного автомобільно-дорожнього університету*

На сьогоднішній день важко уявити компанію без власного представництва в мережі Internet. Як правило таким представництвом стає веб-сайт компанії. За допомогою веб-сайту компанія отримує новий канал залучення клієнтів до своїх товарів або послуг. На сучасному етапі Internet виступає в ролі потужного інструменту з пошуку та надання інформації.

Згідно зі статистикою останніх років, понад 65% жителів України регулярно користуються Internet. Користувачі стають більш вимогливі до наповнення сторінок, зручності навігації та контенту [1].

Розробка веб-сайту дозволяє використовувати сучасні технології для розвитку інформаційної підтримки та реклами кампаній. Великий пласт соціальних груп, які підключаються до мережі Internet в пошуках інформації, весь час розширюється. Процес розробки веб-сайту забезпечить приплив нових користувачів, за допомогою сайту можна домогтися швидкого реагування на побажання користувачів і вносити відповідні зміни в роботу компанії [1, 2].

Проектування сайту – це моделювання ефективного онлайн-ресурсу, первинний етап у розробці сайту і саме він відповідає за адекватність подальшої розробки. Під час проектування сайту, розробляється ідея сайту, визначаються його цілі і завдання, а також способи їх досягнення. По завершенні етапу

проектування, розробники мають чітке уявлення про те, як виглядатиме і працюватиме завершений проект.

Визначення структури сайту містить у собі планування розділів, системи навігації по сайті. Як буде виглядати меню сайту? Де і як будуть розташовані ті або інші модулі? Відповіді на ці питання дозволяють уже на етапі проектування представити, як буде виглядати майбутній сайт.

Етап проектування сайту часто ігнорують і відразу розпочинають створення сайту. Проектування допомагає оптимізувати процес розробки і ефективно розподіляти роботу. Оскільки дизайн, верстка та програмування здійснюються на основі прототипу, краще вносити зміни в попередню модель – це набагато дешевше і заощадить час [2].

Створення концепції сайту.

На етапі розробки концепції сайту необхідно сформулювати ідеологічну базу та узгодити напрямок розвитку з клієнтом. Розробка ідеї сайту інформує про те, чи є єдиним бачення проекту. Отримана концепція повинна точно відповідати на запитання, чи є вона кращою для вирішення поставлених завдань. Враховується також емоційний відгук цільових користувачів.

Проектування структури сайту.

Лояльність значною мірою залежить від того, наскільки проектування графічного інтерфейсу нового сайту відповідна моделі користувачів– їх очікуванням. Тому необхідно встановити такий порядок розташування сторінок сайту один стосовно одного, щоб ця структура була інтуїтивно зрозумілою користувачам. Розробка структури сайту складається з двох рівнів: внутрішній і зовнішній. Внутрішня структура визначає логічний зв'язок між сторінками і залежить від UX проектування дизайну. Він також визначається зовнішньою структурою (навігація сайту).

UI дизайн інтерфейсів: проектування дизайну сайту.

Креативне проектування дизайну веб-сайтів забезпечує приємний і простий досвід людино-машинної взаємодії. UI дизайнер ґрунтується на рекомендаціях UX дизайнера, намагаючись максимально передати легкість побудови роботи з сайтом. На даному етапі емоційний відгук грає головну роль в ухваленні рішення на користь того або іншого дизайну [3].

Задача проектування сайту – максимально детально визначити всі аспекти робіт, створити єдине бачення проекту. Концепція UI/UX– це робота, що виконуються з урахуванням художніх підходів і законів.

Література

1. Катерина Зарембо. Українське покоління Z: цінності та орієнтири. [Електронний ресурс] / Режим доступу:http://neweurope.org.ua/wp-content/uploads/2017/11/Ukr_Generation_ukr_inet-2.pdf
2. Методи проектування. [Електронний ресурс] / Режим доступу: <http://sites.znu.edu.ua/webprog/lect/1239.ukr.html>
3. Проектування веб-сайту. [Електронний ресурс] / Режим доступу: <http://sites.znu.edu.ua/webprog/lect/1194.ukr.html>

ТЕЛЕГРАМ БОТ ДЛЯ МАГАЗИНУ ОДЯГУ

Портніхін К.О.

Харківський національний автомобільно-дорожній університет, Харків

У сучасному світі месенджери стали невід'ємною частиною нашого життя, забезпечуючи швидку та зручну комунікацію. Одним з найпопулярніших месенджерів є Telegram, який надає можливість створення ботів - автоматизованих програм для взаємодії з користувачами. Розробка телеграм-ботів стає все більш актуальною завдяки їхньому широкому застосуванню в бізнесі, освіті, сфері послуг тощо.

Телеграм-боти стали надзвичайно актуальними і корисними в наш час з кількох причин:

1. Автоматизація процесів: Телеграм-боти дозволяють автоматизувати багато рутинних завдань, таких як відповіді на поширені запитання, обробка замовлень, надсилання повідомлень та інше. Це значно спрощує роботу для бізнесів та організацій.

2. Доступність та зручність: Телеграм є однією з найпопулярніших месенджерів у світі, з мільйонами користувачів. Боти в Телеграмі доступні будь-кому, хто має додаток, і можуть взаємодіяти з користувачами в будь-який час і будь-де.

3. Широкий спектр застосувань: Телеграм-боти використовуються в різних сферах, включаючи електронну комерцію, освіту, медицину, розваги, банківську справу та багато іншого. Вони можуть надавати інформацію, проводити опитування, здійснювати платежі, нагадувати про події і навіть навчати.

4. Інтерактивність та персоналізація: Боти можуть забезпечувати інтерактивну та персоналізовану взаємодію з користувачами. Вони можуть запам'ятовувати переваги користувачів і відповідно налаштовувати свій функціонал.

5. Ефективність комунікацій: Використання ботів для масових розсилок, новинних повідомлень або оповіщень дозволяє швидко і ефективно доносити важливу інформацію до великої аудиторії.

6. Інтеграція з іншими сервісами: Боти можуть бути інтегровані з іншими онлайн-сервісами та платформами, такими як CRM-системи, електронні платіжні системи, соціальні мережі та інші, що робить їх ще більш корисними та функціональними.

7. Безпека та конфіденційність: Телеграм відомий своєю увагою до безпеки та конфіденційності користувачів. Це особливо важливо для ботів, які обробляють персональні дані або конфіденційну інформацію.

Загалом, телеграм-боти значно підвищують ефективність і продуктивність у багатьох сферах діяльності, надаючи зручний і доступний інструмент для автоматизації та покращення взаємодії з користувачами.

Вибір бібліотеки для розробки бота

Однією з найпопулярніших бібліотек для створення телеграм-ботів на Python є Aiogram.

Aiogram – це асинхронний Python-фреймворк для розробки ботів для Telegram, який базується на бібліотеці `asuncio`. Aiogram дозволяє легко створювати боти з використанням Telegram Bot API, забезпечуючи простий та інтуїтивно зрозумілий інтерфейс для розробників.

Основні переваги Aiogram:

1. Асинхронність: Використання `asuncio` дозволяє обробляти багато запитів одночасно, що підвищує ефективність роботи бота.
2. Простота використання: Aiogram має простий і зрозумілий інтерфейс, що дозволяє швидко створювати боти.
3. Розширюваність: Aiogram легко розширюється за допомогою додаткових модулів та плагінів.
4. Повна підтримка Telegram Bot API: Aiogram підтримує всі можливості Telegram Bot API, що дозволяє створювати боти з широким функціоналом.

Опис основних модулів

Основні модулі включають:

- Модуль для взаємодії з Telegram API.
- Модуль для обробки команд користувачів.
- Модуль для взаємодії з базою даних.

BotFather - це спеціальний бот в Telegram, який допомагає користувачам створювати та налаштовувати своїх власних ботів. Ось деякі основні функції і використання BotFather:

1. Створення ботів: BotFather дозволяє користувачам створювати нових ботів, надаючи їм імена та унікальні ідентифікатори (юзернейми).
2. Отримання API токенів: Після створення бота BotFather генерує унікальний API токен, який потрібен для доступу до Telegram API і управління ботом.
3. Налаштування параметрів бота: BotFather дозволяє користувачам налаштовувати різні параметри бота, такі як опис, профільна картинка, команди, доступ до груп і багато іншого.
4. Управління ботами: Користувачі можуть використовувати BotFather для редагування параметрів своїх існуючих ботів, наприклад, зміни юзернейму чи профільної інформації.
5. Доступ до документації: BotFather надає користувачам доступ до документації Telegram Bot API, де вони можуть знайти інформацію про всі можливості та функції, які доступні для розробки ботів.

Загалом, BotFather робить процес створення і управління ботами в Telegram більш простим і зручним для користувачів, надаючи доступ до необхідних інструментів і функцій через один зручний інтерфейс.

Реалізація телеграм-бота

1. Імпортуються всі необхідні модулі.

2. Оголошується асинхронна функція main, яка створює об'єкти Bot та Dispatcher, підключає маршрутизатор та запускає опитування для обробки повідомлень.

3. Перевіряється, чи скрипт виконується як основний модуль.

4. Виконується функція main за допомогою asyncio.run().

5. При натисканні Ctrl+C виконання переривається, виводиться повідомлення "Бот вимкнено".

Дамо детальний опис кожного елемента:

1. `Router`: Цей клас відповідає за маршрутизацію повідомлень та запитів від користувачів до відповідних обробників.

2. Словник `item_translation`: Це словник, який містить переклад назв товарів на правильну форму та їх ціни.

3. Стани `Register` та `Order`: Ці класи визначають різні стани, через які проходить користувач при реєстрації та оформленні замовлення.

4. Обробники команд та повідомлень:

- `cmd_start`, `cmd_help`: Обробляють команди `/start` та `/help`, які викликаються користувачами.

- `catalog`, `contacts`, `about_us`: Відображають каталог товарів, контактну інформацію магазину та інформацію про магазин відповідно.

- `t_shirt`, `sneakers`, `cap`: Обробляють вибір категорії товарів та відображають підкатегорії.

- `add_to_cart`, `remove_from_cart`: Додають або видаляють товари з кошика користувача.

- `show_cart`: Відображає вміст кошика користувача.

- `start_order`, `order_surname`, `order_name`, `order_phone`, `order_city`, `order_nova_poshta`: Обробляють процес оформлення замовлення.

- `register`, `register_name`, `register_age`, `register_number`: Обробляють процес реєстрації користувача.

5. Клавіатури:

- `main`, `catalog`, `sneakers`, `t_shirts`, `caps`, `contacts`, `get_number`: Ці клавіатури допомагають користувачам навігувати та взаємодіяти з ботом шляхом вибору пунктів меню або категорій товарів.

Цей код створює повнофункціональний телеграм-бот, який може взаємодіяти з користувачами, допомагати їм з реєстрацією, вибором товарів та оформленням замовлень.

Тестування телеграм бота

Під час тестування бота було помічено кілька помилок, які було виправлено:

1. Не коректно відображалася ціна товару для категорії "кепки". Помилка полягала в тому, що ключі в словнику `item_translation` були вказані в нижньому регістрі, але в обробнику `add_to_cart` використовувався регістр з великої

літери. Це було виправлено шляхом зміни ключів на великі літери у словнику `item_translation`.

2. Під час тестування виявлено, що кнопка "про нас" не активує відповідну функцію. Це було виправлено шляхом додавання функції, що відповідає за обробку цієї команди.

3. Додано функціонал для реєстрації користувачів. Тепер користувачі можуть ввести свої дані: ім'я, вік та номер телефону.

4. Додано клавіатуру для зручного введення номера телефону за допомогою кнопки "Відправ номер".

Після виправлення цих помилок бот працює коректно та має розширений функціонал.

Приклад роботи телеграм бота

При запуску бот вітає користувача з повідомленням "Привіт!" та відображає головне меню з варіантами:

1. "Каталог" - для перегляду товарів у каталозі.
2. "Корзина" - для перегляду вмісту корзини та оформлення замовлення.
3. "Контакти" - для отримання контактної інформації про магазин.
4. "Про нас" - для отримання інформації про магазин.

Також бот запитує користувача, як справи, щоб створити зручну атмосферу для спілкування.

Коли користувач натискає на будь-який розділ, наприклад, "Каталог", бот показує відповідний розділ з доступними категоріями товарів. Наприклад, для "Каталогу" це можуть бути категорії "Футболки", "Кросівки" та "Кепки". Кожна категорія представлена як кнопка, на яку користувач може натиснути для перегляду товарів у цій категорії.

Користувач може замовити товар, виконавши такі кроки:

1. Вибір товару: Користувач переглядає доступні товари у каталозі та обирає той, який бажає придбати.

2. Додавання до корзини: Після вибору товару користувач натискає кнопку "Додати до корзини", щоб покласти товар до своєї корзини покупок.

3. Перегляд корзини: Користувач переходить у розділ "Корзина", де може переглянути список вибраних товарів та їх загальну суму.

4. Оформлення замовлення: Після перегляду корзини, користувач натискає кнопку "Замовити", щоб перейти до оформлення замовлення.

5. Введення даних: Користувач вводить необхідні дані для оформлення замовлення, такі як ім'я, прізвище, номер телефону, місто та номер відділення Нової Пошти.

6. Підтвердження замовлення: Після введення всіх необхідних даних користувач підтверджує замовлення, після чого отримує підтвердження замовлення з вказаними деталями та загальною сумою.

Висновки

Під час тестування бота виявлено деякі помилки, але завдяки вдосконаленням і виправленням коду, бот працює стабільно і безперебійно. Він успішно виконує свою роботу, дозволяючи користувачам переглядати каталог товарів, додавати їх до корзини, оформляти замовлення та отримувати підтвердження замовлення з деталями. Розділ "Каталог" надає зручний спосіб перегляду доступних товарів, розділених за категоріями. При натисканні на певну категорію, користувач може побачити варіанти товарів у цій категорії та додати їх до корзини.

У розділі "Корзина" користувач може переглянути товари, які він додав до корзини, та їх загальну суму. Крім того, в цьому розділі є можливість видалити товари з корзини або зробити замовлення. Під час замовлення користувач вводить необхідні дані, такі як ім'я, прізвище, номер телефону, місто та номер відділення Нової Пошти. Після підтвердження замовлення, користувач отримує підтвердження з деталями замовлення та загальною сумою.

В цілому, бот пропонує зручний і простий у використанні інтерфейс для покупців та дозволяє їм легко здійснювати покупки.

Список літератури

1. Googletrans: Free and Unlimited Google translate API for Python [Електронний ресурс]. Режим доступу: <https://py-googletrans.readthedocs.io/en/latest/>. Дата доступу: 15.04.2024
2. aiogram 3.7.0 documentation [Електронний ресурс]. Режим доступу: <https://docs.aiogram.dev/en/latest/>. Дата доступу: 15.04.2024
3. OpenWeatherMap API documentation [Електронний ресурс]. Режим доступу: <https://openweathermap.org/api>. Дата доступу: 15.04.2024
4. Rest Countries API documentation [Електронний ресурс]. Режим доступу: <https://restcountries.com>. Дата доступу: 15.04.2024

КІБЕРБЕЗПЕКА У ХМАРНИХ ТЕХНОЛОГІЯХ

Балабай А.О., Керівник Плехова Г.А.

Харківський національний автомобільно-дорожній університет, Харків.

Вступ

В сучасному цифровому світі хмарні технології є визначальною складовою частиною інформаційної інфраструктури, дозволяючи користувачам та компаніям зберігати, обробляти та ділитися даними за допомогою віртуальних ресурсів через Інтернет. Це привело до стрімкого росту популярності хмарних технологій, оскільки вони надають низку переваг, таких як гнучкість, масштабованість та зменшення витрат. Проте, разом із зростанням використання хмарних сервісів зростає і ризик порушення кібербезпеки. У цій статті ми розглянемо ключові виклики, які ставлять під загрозу безпеку в хмарних середовищах, а також запропонуємо стратегії захисту для ефективного контролю за цими ризиками.

Зростаюча популярність та розширення застосування хмарних технологій
Хмарні технології стали ключовим елементом сучасного ІТ-пейзажу, забезпечуючи користувачам миттєвий доступ до потужних обчислювальних ресурсів та інфраструктури за потреби. Замість власних серверів і обладнання, компанії можуть легко використовувати хмарні ресурси для зберігання даних, розгортання програмного забезпечення та виконання обчислень в реальному часі. Це дозволяє їм підтримувати високий рівень гнучкості та масштабованості, не залежно від обсягу роботи або потреб користувачів.

Загрози кібербезпеки в хмарних середовищах

Однак разом із зростанням використання хмарних технологій зростає і кількість загроз для кібербезпеки. Несанкціонований доступ до даних, атаки на відмову в обслуговуванні (DoS та DDoS), маніпулювання даними та вразливості віртуалізації - це лише кілька з ключових загроз, які можуть виникнути в хмарних середовищах. Неспроможність адекватно захистити дані може призвести до небажаних наслідків, таких як втрата конфіденційної інформації, порушення приватності користувачів та серйозні фінансові втрати.

Стратегії захисту в хмарних середовищах

Забезпечення кібербезпеки в хмарних технологіях вимагає комплексного підходу та ефективного управління ризиками. Ось декілька стратегій, які можуть допомогти у забезпеченні безпеки в хмарних середовищах:

Шифрування даних: Використання сильного шифрування для захисту конфіденційної інформації в хмарних середовищах. Це може включати застосування алгоритмів шифрування з використанням ключів високої довжини, таких як AES-256, а також регулярне оновлення ключів шифрування для забезпечення максимальної безпеки.

Постійний моніторинг безпеки: Регулярний моніторинг активності та виявлення потенційних загроз для швидкого реагування. Це може включати використання систем моніторингу заходів безпеки, які автоматично аналізують великі обсяги даних для виявлення аномальної активності та надсилають сповіщення про можливі загрози.

Двофакторна аутентифікація: Використання додаткового рівня аутентифікації для запобігання несанкціонованому доступу до хмарних ресурсів. Крім стандартного введення пароля, користувачам може бути запропоновано використовувати додатковий метод аутентифікації, такий як OTP (одноразовий пароль) або біометричні дані.

Регулярні оновлення та патчі: Забезпечення актуальності та безпеки систем за допомогою регулярних оновлень та патчів. Організації повинні слідкувати за випуском нових версій програмного забезпечення та оперативно встановлювати патчі для виправлення виявлених вразливостей.

Захист мережі: Встановлення та підтримання ефективних мережевих заходів безпеки, таких як брандмауери та інтерфейсні заходи безпеки. Це може

включати встановлення правил фільтрації пакетів, моніторинг мережевої активності та блокування небезпечного трафіку.

Навчання та освіта користувачів: Проведення навчання та наведення прикладів щодо безпеки в ІТ-сфері для підвищення обізнаності користувачів щодо потенційних загроз. Організації можуть проводити тренінги та семінари з питань кібербезпеки, а також надавати доступ до навчальних матеріалів та ресурсів онлайн.

Приклад ефективності

Регулярні оновлення та патчі вважаються однією з ключових стратегій захисту в хмарних середовищах. Ця стратегія допомагає уникнути багатьох потенційних загроз, забезпечуючи актуальність та безпеку систем та даних.

Один з прикладів успішного використання стратегії регулярних оновлень та патчів - це казус компанії X, яка використовує хмарні сервіси для зберігання та обробки конфіденційних даних клієнтів. Компанія X регулярно виконує оновлення свого програмного забезпечення та встановлює патчі, які випускають розробники.

У лютому цього року компанія X виявила вразливість в одному зі сторонніх додатків, які використовувалися для обробки даних. Завдяки системі регулярних оновлень, компанія X миттєво отримала повідомлення від постачальника хмарних послуг про випуск патча для виправлення цієї вразливості. Компанія X негайно встановила патч і вжила заходів для мінімізації ризиків. Завдяки швидкій реакції та своєчасному встановленню патчів, компанія X уникнула можливих наслідків вразливості, таких як втрата даних клієнтів або порушення їх конфіденційності. Цей приклад підтверджує ефективність стратегії регулярних оновлень та патчів у забезпеченні безпеки в хмарних середовищах.

Також прикладом ефективності цієї стратегії є кібератака WannaCry, яка відбулася в 2017 році. Уразливість, яку використовувала ця атака, була відома як "EternalBlue" і стосувалася операційних систем Windows. Компанія Microsoft випустила патч для цієї уразливості за місяць до атаки, однак багато організацій не встигли вчасно встановити цей патч на своїх системах. Ті, хто зробив це, були захищені від атаки, в той час як ті, хто не оновив свої системи, стали жертвами вірусу WannaCry.

Висновок

Захист кібербезпеки в хмарних технологіях є складною, але критично важливою задачею в сучасному цифровому світі. Неспроможність забезпечити ефективний захист може призвести до серйозних наслідків, таких як втрата даних, порушення приватності та фінансові втрати. З цим у свідомості, організації повинні приділити належну увагу кібербезпеці та вжити всіх можливих заходів для захисту своєї інфраструктури та даних. Сукупність розглянутих стратегій може допомогти організаціям ефективно захищатися в хмарних середовищах і забезпечувати безпеку своїх даних та інфраструктури у цифровому віці.

UDC 004.773

**APPLYING OF BIG DATA ANALYTICS IN INTELLIGENT
TRANSPORT SYSTEMS***Babenko V.O., Drozdyk Ye.V.**Kharkiv National Automobile and Highway University, Kharkiv*

There are several basic and advanced types of data structures, all designed to arrange data to suit a specific purpose. Data structures make it easy for users to access and work with the data they need in appropriate ways. Most importantly, data structures frame the organization of information so that machines and humans can better understand it.

In computer science and computer programming, a data structure may be selected or designed to store data for the purpose of using it with various algorithms. In some cases, the algorithm's basic operations are tightly coupled to the data structure's design. Each data structure contains information about the data values, relationships between the data and in some cases functions that can be applied to the data.

For instance, in an object-oriented programming language, the data structure and its associated methods are bound together as part of a class definition. In non-object-oriented languages, there may be functions defined to work with the data structure, but they are not technically part of the data structure.

Software engineers use algorithms that are tightly coupled with the data structures such as lists, queues and mappings from one set of values to another. This approach can be fused in a variety of applications, including managing collections of records in a relational database and creating an index of those records using a data structure called a binary tree.

Some examples of how data structures are used include the following:

Storing data. Data structures are used for efficient data persistence, such as specifying the collection of attributes and corresponding structures used to store records in a database management system.

Managing resources and services. Core operating system (OS) resources and services are enabled through the use of data structures such as linked lists for memory allocation, file directory management and file structure trees, as well as process scheduling queues.

Data exchange. Data structures define the organization of information shared between applications, such as TCP/IP packets.

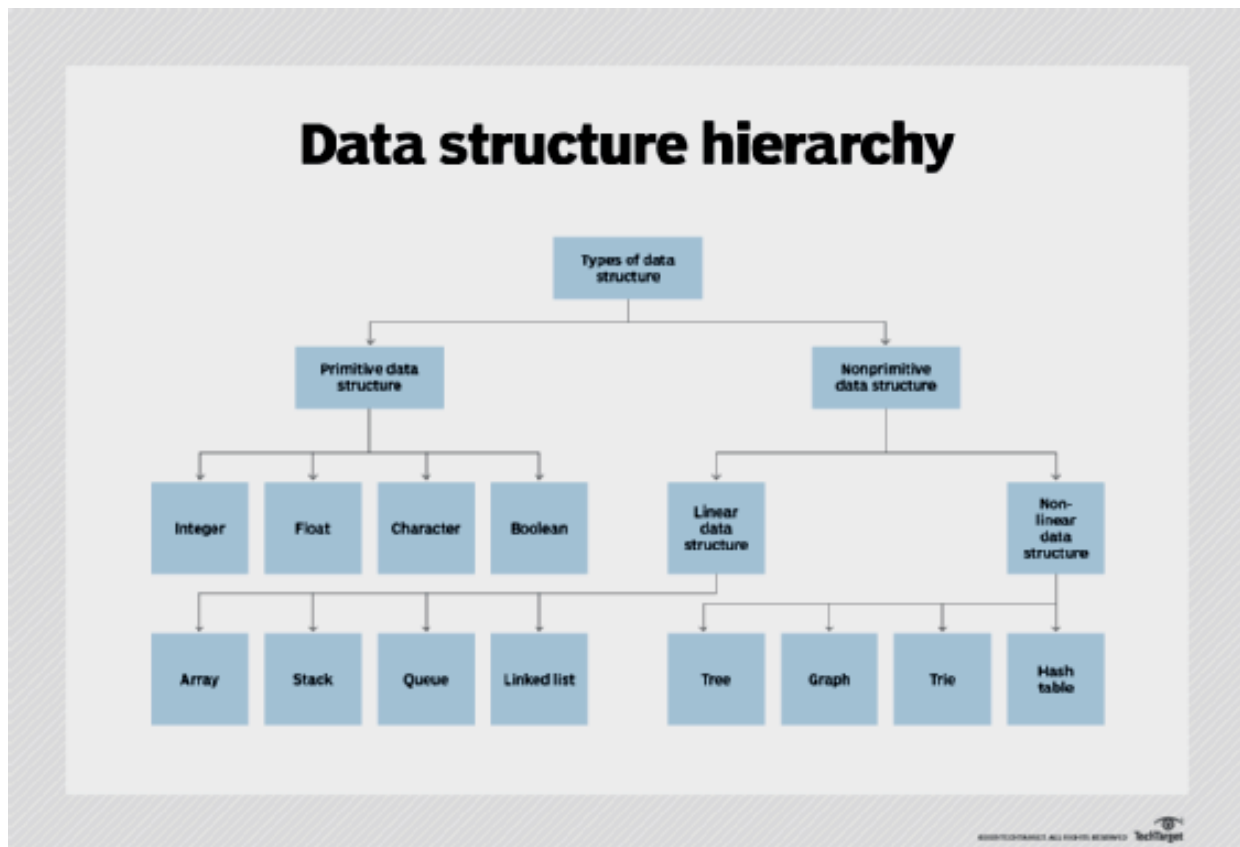


Figure 1 – Data structure design

Ordering and sorting. Data structures such as binary search trees – also known as an ordered or sorted binary tree – provide efficient methods of sorting objects, such as character strings used as tags. With data structures such as priority queues, programmers can manage items organized according to a specific priority. Indexing even more sophisticated data structures such as B-trees are used to index objects, such as those stored in a database.

Searching. Indexes created using binary search trees, B-trees or hash tables speed the ability to find a specific sought-after item. Scalability. Big data applications use data structures for allocating and managing data storage across distributed storage locations, ensuring scalability and performance. Certain big data programming environments – such as Apache Spark – provide data structures that mirror the underlying structure of database records to simplify querying.

As more organizations adopt big data platforms, concern mounts that application development may suffer from the lack of good practices for managing the data powering those applications. When we talk about big data management in relation to big data platforms (like those combining commodity hardware with Hadoop), it's clear that big data technologies have created a need for new and different data management tools and processes.

This is the crux of the issue. Any big data management strategy must include technology to support stream processing that scans, filters and selects the meaningful information for capture, storage and subsequent access.

Big Data Management in Automobile Industry

This slide explains the big data use cases in the automobile industry and how it would help automobile firms in supply chain management, connected cars, automobile financing, predictive analysis, and design and production.



Figure 2 – Example of Big Data management in Automobile industry

Managing big data not only subsumes many of the conventional approaches to data modeling and architecture, it entails a new cadre of technologies and processes to enable broader data accessibility and usability. A big data management strategy must embrace tools enabling data discovery, data preparation, self-service data accessibility, collaborative semantic metadata management, data standardization and cleansing, and stream processing engines. Being aware of these implications can dramatically speed the time-to-value of your big data program.

The phenomenon of Big Data exacerbates the tension between potential benefits and privacy risks by upping the ante on both sides of the equation. Any project can fail for any number of reasons: bad management, poor budget management, or just a lack of relevant skills. However, big data projects bring their own specific risks. Disturbingly Currently Only 13% of Companies Achieve Full-scale Implementation of Their In-house Big Data Projects

Such a low success rate should be concerning for organizations embarking on big data projects since many businesses are choosing to adopt big data without a clear understanding of what the return on investment (ROI) will be.

Big data enables the designers to develop web applications that are more engaging and effective in delivering useful information to the users.

As more and more businesses continue to rely on the internet to promote their services and products, big data can help their manager to make better decisions. However, they require the right tools and skills to analyze and understand the data.

Our comprehension from the program that we could build any application in our computer but it require us to have data designed well beside the structure of the application.

Reference:

1. N. Carr, *The Big Switch: Rewiring the World, from Edison to Google*. W. W. Norton & Co., New York, 2020.
2. The Research and Application of Network Teaching Platform Based on Cloud Computing, Z. Tao and J. Long, *International Journal of Information and Education Technology*, Vol. 1, No. 3, August 2011
3. <https://www.selecthub.com/big-data-analytics/types-of-big-data-analytics/>

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПОБУДОВИ ЛОГІСТИЧНИХ ЛАНЦЮГІВ

Барашков Владислав Сергійович, Пронін С.В.

Харківський національний автомобільно-дорожній університет

У сучасному світі інформаційні технології відіграють ключову роль у багатьох сферах діяльності, зокрема в логістиці. Одним з найбільш ефективних інструментів для побудови та управління логістичними ланцюгами є графові бази даних, такі як Neo4j. Ця стаття присвячена використанню Neo4j для побудови логістичних ланцюгів, опису її основних функціональних можливостей та переваг, а також прикладу реального застосування.

Інформаційні технології у логістиці

Логістика – це комплексний процес планування, реалізації та контролю ефективного та економічного переміщення і зберігання товарів, послуг і інформації від точки походження до точки споживання. Логістичні ланцюги, або ланцюги постачань, є основою цього процесу, забезпечуючи взаємодію між постачальниками, виробниками, дистриб'юторами та кінцевими споживачами.

Логістичні ланцюги включають різні етапи, такі як закупівля сировини, виробництво, зберігання, транспортування та доставка готової продукції. Ефективне управління цими етапами вимагає точного планування, координації та контролю, що можна досягти завдяки використанню сучасних інформаційних технологій.

Використання інформаційних технологій у логістиці дозволяє оптимізувати процеси, підвищити ефективність управління та знизити витрати. Основні напрями використання ІТ у логістиці включають:

1. Автоматизація процесів: Використання спеціалізованого програмного забезпечення для автоматизації логістичних операцій, таких як складування, обробка замовлень та управління запасами.

2. Системи управління ланцюгами постачань (SCM): Інтеграція інформаційних систем, що дозволяє ефективно планувати, виконувати та контролювати логістичні операції на всіх етапах ланцюга постачань.

3. Графові бази даних: Використання графових баз даних для моделювання та аналізу складних логістичних ланцюгів, що включають численні взаємозв'язки між різними елементами системи.

Використання Neo4j у логістиці

Neo4j – це графова база даних, яка дозволяє зберігати та обробляти дані у вигляді графів, де вузли представляють об'єкти, а ребра – взаємозв'язки між ними. Це робить Neo4j ідеальним інструментом для моделювання складних логістичних ланцюгів, що включають велику кількість елементів та зв'язків між ними.

1. Гнучка модель даних: Neo4j дозволяє зберігати дані у вигляді графів, що забезпечує високу гнучкість у моделюванні складних систем. Це дозволяє легко додавати нові елементи та зв'язки без необхідності зміни структури бази даних.

2. Висока продуктивність: Neo4j оптимізований для роботи з графами, що забезпечує швидке виконання запитів та обробку великих обсягів даних. Це особливо важливо для логістики, де необхідно швидко аналізувати та обробляти інформацію для прийняття рішень.

3. Інтеграція з іншими інструментами: Neo4j підтримує інтеграцію з різними інструментами та плагінами, такими як APOC (Awesome Procedures on Cypher) та Graph Data Science Library, що дозволяє розширювати функціональні можливості бази даних.

4. Візуалізація даних: Neo4j надає потужні інструменти для візуалізації графів, що дозволяє легко аналізувати складні логістичні ланцюги та виявляти можливості для оптимізації.

Розглянемо приклад використання Neo4j для управління логістичними ланцюгами в компанії, що займається виробництвом та дистрибуцією товарів. Компанія може використовувати Neo4j для моделювання всього ланцюга постачань, від постачальників сировини до кінцевих споживачів.

1. Моделювання ланцюга постачань: У Neo4j створюються вузли для кожного елемента ланцюга (постачальники, виробничі підрозділи, склади, дистриб'ютори та споживачі) та ребра для зв'язків між ними. Це дозволяє побудувати повну картину логістичної системи.

2. Аналіз та оптимізація: Використовуючи можливості Neo4j, компанія може аналізувати ланцюг постачань для виявлення вузьких місць, оптимізації маршрутів транспортування та зменшення витрат. Наприклад, можна швидко визначити найбільш ефективні шляхи доставки товарів від виробника до споживача.

3. Моніторинг та управління: Neo4j дозволяє інтегрувати дані в режимі реального часу для моніторингу стану ланцюга постачань. Це забезпечує можливість оперативного реагування на зміни та швидке прийняття рішень.

Висновок

Використання графових баз даних, таких як Neo4j, для побудови та управління логістичними ланцюгами надає значні переваги у гнучкості моделювання, продуктивності та можливостях аналізу. Це дозволяє компаніям оптимізувати свої логістичні процеси, знижувати витрати та підвищувати ефективність управління. У майбутньому, з розвитком технологій та збільшенням обсягів даних, використання таких інструментів ставатиме все більш актуальним та необхідним для успішної діяльності в умовах зростаючої конкуренції.

Список літератури

1. The Neo4j Graph Data Science Library Manual v2.6 [електронний ресурс]. <https://neo4j.com/docs/graph-data-science/current/> (Режим доступу до 21.05.2024)
2. Basic workflow Neo4j [електронний ресурс]. <https://neo4j.com/docs/graph-data-science/current/getting-started/basic-workflow/> (Режим доступу до 21.05.2024)
3. End-to-end workflow Neo4j [електронний ресурс]. <https://neo4j.com/docs/graph-data-science/current/getting-started/fastrp-knn-example/> (Режим доступу до 21.05.2024)
4. Machine learning pipeline [електронний ресурс]. <https://neo4j.com/docs/graph-data-science/current/getting-started/ml-pipeline/> (Режим доступу до 21.05.2024)
5. Cypher Cheat Sheet [електронний ресурс]. https://neo4j.com/docs/cypher-cheat-sheet/5/neo4j-community#_delete (Режим доступу до 21.05.2024)

УДК 656.1+681.5+004.04

ПЕРСПЕКТИВИ ВПРОВАДЖЕННЯ ПРОГНОСТИЧНОГО ОБСЛУГОВУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Грицук Валерій Юрійович, аспірант, Пронін С.В.

Харківський національний автомобільно-дорожній університет

Анотація. В роботі розглянута можливість використання даних, отриманих за допомогою OBD в системній взаємодії прогностичного обслуговування транспортних засобів. Представлено варіанти збору таких даних до бази даних та можливості використання даних для машинного навчання.

Ключові слова: системна взаємодія, база даних, транспортний засіб, передача даних, обробка даних, навчання, машинне навчання, OBD

Прогностичне обслуговування транспортних засобів актуальне для України, оскільки це може допомогти зменшити витрати на ремонт та збільшити тривалість служби компонентів. Воно також може підвищити безпеку на дорогах, виявляючи потенційні проблеми з автомобілями на ранніх стадіях. Крім того, використання великих даних та машинного навчання в рамках

прогностичного обслуговування може допомогти автомобільним компаніям отримувати цінні висновки про стан автомобіля, ефективність палива, водійську поведінку та інше.

В системній взаємодії в прогностичному обслуговуванні транспортних засобів (рис. 1) можна виділити основні блоки (збір і передавання даних, зберігання та обробка даних, аналіз і використання).

Блок центру накопичення та обробки даних може становити собою реляційну базу даних (або бути елементом більш складної структури), які найбільш часто [1] використовуються для рішення різноманітних задач (рис. 2), в тому числі і в автомобільному транспорті [2].

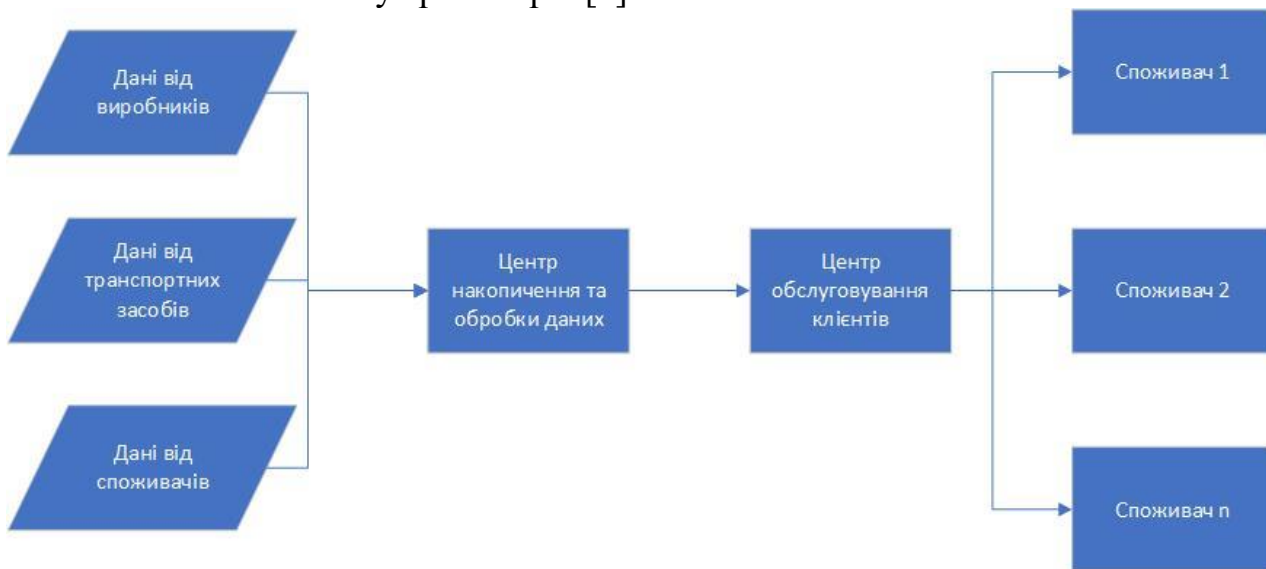
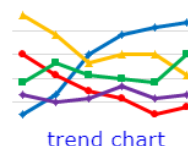


Рис. 1. Системна взаємодія прогностичного обслуговування

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



419 systems in ranking, April 2024

Rank			DBMS	Database Model	Score		
Apr 2024	Mar 2024	Apr 2023			Apr 2024	Mar 2024	Apr 2023
1.	1.	1.	Oracle +	Relational, Multi-model	1234.27	+13.21	+5.99
2.	2.	2.	MySQL +	Relational, Multi-model	1087.72	-13.77	-70.06
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	829.80	-16.01	-88.73
4.	4.	4.	PostgreSQL +	Relational, Multi-model	645.05	+10.15	+36.64
5.	5.	5.	MongoDB +	Document, Multi-model	423.96	-0.57	-17.93
6.	6.	6.	Redis +	Key-value, Multi-model	156.44	-0.56	-17.11
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model	134.78	-0.01	-6.29
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model	127.49	-0.26	-18.00
9.	9.	↑ 12.	Snowflake +	Relational	123.20	-2.18	+12.07
10.	10.	↓ 9.	SQLite +	Relational	116.01	-2.15	-18.53

Рис. 2. Статистика використання баз даних

Їх перевагами є структурованість, відносини між даними, масштабованість, універсальність та використання запитів SQL.

Обмін даними між джерелами даних (транспортні засоби, споживачі, виробники) з серверною частиною блоку центру накопичення та обробки даних може здійснюватися з використанням технології 4G або 5G (принцип OBD III або RTM для Китаю) [3]. З іншого боку це може бути набір даних (як приклад можна навести датасет, підготовлений Національною адміністрацією безпеки дорожнього руху США (NHTSA) та розміщений на платформі конкурсу з науки про дані Kaggle (рис. 3) [4]).

Record ID	NHTSA Campaign	Manufacturer Ca...	Vehicle Make	Vehicle Model	# Model
1	20684 unique values	[null] 66% 215-D 0% Other (42023) 34%	FORD 5% CHEVROLET 4% Other (112820) 91%	15164 unique values	1949
1	02V288000	02S41	FORD	FOCUS	2001
2	02V288000	02S41	FORD	FOCUS	2000
3	02V236000		JAYCO	FT EAGLE 10 SG	2003
4	02V237000		HOLIDAY RAMBLER	ADMIRAL SE	2003
5	02V237000		HOLIDAY RAMBLER	ADMIRAL SE	2002
6	02V237000		HOLIDAY RAMBLER	ADMIRAL	1999
7	02V237000		MONACO COACH	MONARCH SE	2003
8	02V237000		MONACO COACH	MONARCH SE	2002
9	02V237000		MONACO COACH	LAPALMA	1999
10	02V237000		MONACO COACH	LAPALMA	2003

Рис. 3. Фрагмент датасету, підготовленого Національною адміністрацією безпеки дорожнього руху США (NHTSA)

Взагалі, доволі часто, в аналізі даних використовуються саме датасети (структуровані набори даних, представлені у вигляді таблиці). Кожен стовпець в таблиці представляє певну ознаку (або змінну), а кожен рядок відповідає конкретному спостереженню (або запису) [5]. Датасети можуть включати різні типи даних, такі як специфікації автомобілів, записи про обслуговування, історію автомобілів, дані про споживання палива, дані телеметрії та інше. Ці дані можуть бути використані для дослідження автомобілів, аналізу ринку, прогностичного обслуговування, страхування, управління автопарком та інших цілей. Датасети можуть бути великими або малими, простими або складними, і вони можуть містити дані з різних джерел.

В роботах [6, 7] як один з подальших напрямків розвитку і вдосконалення інтелектуальної транспортної системи розглядається впровадження в їх склад

сервісу дистанційного контролю параметрів технічних систем транспортних засобів (рис. 4), який є одним з напрямків автомобільної телематики поряд з системами навігації, системами контролю трафіку, системами комунікації автомобілів один з одним і інформаційно-командними системами [6 - 8]. Як головний елемент такого сервісу автори розглядають вбудовану діагностичну систему (OnBoard Diagnostics, OBD), що представляє собою сукупність електронних блоків управління, датчиків і виконавчих пристроїв, підключення до автомобільної шини даних.

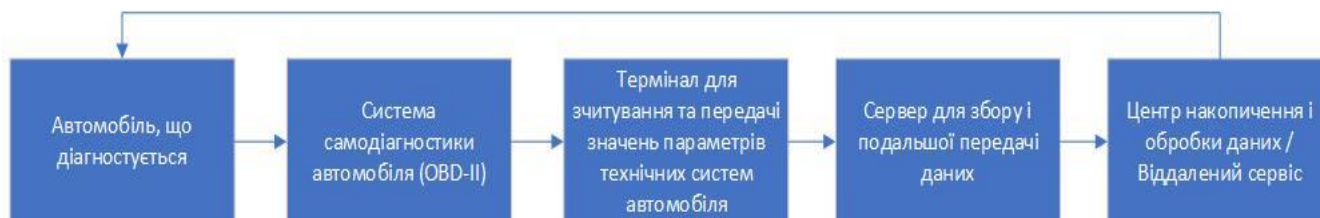


Рис. 4. Схема взаємодії елементів сервісу віддаленого контролю параметрів технічних систем транспортних засобів (за [6], трансформована)

OBD-сканер дозволяє отримувати доступу до багатьох типів даних, включаючи [9]:

- Параметри в реальному часі: оберти двигуна (RPM), швидкість, положення педалі, кут передчасного запалювання, швидкість повітряного потоку, температура охолоджуючої рідини тощо;
- Стан індикатора «Check Engine»;
- Стан готовності до емісії;
- Заморожена рамка: «знімок» параметрів на момент виникнення несправності;
- Діагностичні коди несправностей (DTC);
- Результати перевірки кисневого датчика;
- Ідентифікаційний номер автомобіля (VIN);
- Кількість циклів запалювання.

Ці дані можна розглядати як елемент Big Data в контексті того, що кожен автомобіль генерує велику кількість різноманітних даних, які збираються в режимі реального часу. Відповідно, отримані з OBD дані можуть бути використані для машинного навчання на декількох рівнях:

1. Дані з OBD можуть бути використані для прогнозування потреби в технічному обслуговуванні автомобіля. Наприклад, можна створити моделі, які прогнозують «виклик до дії» на основі кодів DTC та заморожених кадрів [10].

2. Дані з OBD можуть бути використані для аналізу та класифікації стилю водіння. Наприклад, можна зібрати критичні дані про продуктивність автомобіля, включаючи швидкість, оберти двигуна, положення педалі, визначене навантаження двигуна та більше 50 інших параметрів, які можуть бути використані для побудови моделі, яка класифікує поведінку водія та може бути використана для надання зворотного зв'язку для покращення звичок водіння [11, 12].

3. Дані з OBD можуть використовуватися для прогнозування траєкторії автомобіля. Наприклад, можна використовувати модель Long-Short-Term Memory (LSTM) для прогнозування швидкості та курсу на наступну 1 секунду [13].

Таким чином, можна побачити, що дані, отримані за допомогою OBD можуть бути ефективно використані в системній взаємодії прогностичного обслуговування транспортних засобів, в тому числі як компонент Big Data, а також для машинного навчання.

Література

1. DB-Engines Ranking. URL: <https://db-engines.com/en/ranking> (дата звернення 12.04.2024).
2. Грицук А.І., Грицук В.Ю., Ченцов А.В., Матейчик П.В., Краснокутська З.І. Використання інформаційних баз даних в задачах моніторингу автомобільного транспорту. *Сучасні енергетичні установки на транспорті і технології та обладнання для їх обслуговування*. 9-а Міжнародна науково-практична конференція, 13-14 вересня 2018 р. Херсон: ХДМА. 376 с. С.37-39.
3. Nagy, J., Lakatos, I. The past, the present and the future of online road vehicle diagnosis from car and scan tool communication perspective (2023) *Advances in Transdisciplinary Engineering*, 39, pp. 52-58.
4. NHTSA. Vehicle and Tire Recalls, 1967-Present. URL: <https://www.kaggle.com/nhtsa/safety-recalls> (дата звернення 12.04.2024)
5. What is a Dataset: Types, Features, and Examples. URL: <https://www.geeksforgeeks.org/what-is-dataset/> (дата звернення 12.04.2024)
6. Gritsuk, I., Volkov, V., Mateichyk, V., Grytsuk, Y. et al., Information Model of V2I System of the Vehicle Technical Condition Remote Monitoring and Control in Operation Conditions, SAE Technical Paper 2018-01-0024, 2018, <https://doi.org/10.4271/2018-01-0024>.
7. Волков В.П., Грицук І.В., Грицук Ю.В., Волков Ю.В., Володарець М.В. Інформаційні системи моніторингу технічного стану автомобілів. Харків: ХНАДУ, 2018. 300 с.
8. Катунин А.А., Пешехонов М.В. Дистанционный контроль параметров технических систем автомобилей в составе ИТС. *Автомобиль и электроника. Современные технологии*. № 2(5). Харьков: ХНАДУ, 2013. С.114-118.
9. What data is available from OBD? URL: <https://www.obdsol.com/knowledgebase/on-board-diagnostics/what-data-is-available-from-obd/> (дата звернення 12.04.2024)
10. Predicting Vehicle Maintenance using on-board and off-board data URL: <https://ecosystem.evolve-h2020.eu/predicting-vehicle-maintenance-using-on-board-and-off-board-data/> (дата звернення 12.04.2024)
11. Kumar, R., Jain, A. Driving behavior analysis and classification by vehicle OBD data using machine learning. *J Supercomput* 79, 18800–18819 (2023). <https://doi.org/10.1007/s11227-023-05364-3>
12. Peppes N, Alexakis T, Adamopoulou E, Demestichas K. Driving Behaviour Analysis Using Machine and Deep Learning Methods for Continuous Streams of Vehicular Data. *Sensors*. 2021; 21(14):4704. <https://doi.org/10.3390/s21144704>
13. Navali, N., Vanajakshi, L., Bullock, D.M. (2023). Application of On-Board Diagnostics (OBD) Data for Vehicle Trajectory Prediction. In: Devi, L., Errampalli, M., Maji, A., Ramadurai, G. (eds) *Proceedings of the Sixth International Conference of Transportation Research Group of India . CTRG 2021. Lecture Notes in Civil Engineering*, vol 273. Springer, Singapore. https://doi.org/10.1007/978-981-19-4204-4_19

АНАЛІЗ СТАНДАРТНОЇ АВТОМАТИЗАЦІЇ БІЗНЕС ПРОЦЕСІВ ПІДПРИЄМСТВ ТА АВТОМАТИЗАЦІЇ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

*Петренко Ю.А., д.т.н., професор, Жабін О.Ю., аспірант
Харківський національний автомобільно-дорожній університет*

Анотація. Використання штучного інтелекту в автоматизації бізнес процесів є одним з стратегічних напрямків розвитку цифрової трансформації сучасного бізнесу. Потенціал використання штучного інтелекту в автоматизації бізнес процесів може бути застосований в різних напрямках бізнесу, як то сфера виробництва так і у сферах соціальних послуг та обслуговування.

Ключові слова: автоматизація бізнес процесів, штучний інтелект, когнітивна автоматизація, інтелектуальна процесна автоматизація.

У сучасному світі бізнес все частіше шукає можливість фундаментальної цифрової трансформації свої процесів, де системи можуть стати інтелектуальними та такими, що самостійно адаптуються до динаміки бізнесу, що постійно змінюється, та чутливих уподобань покупців [1,2]. Крім того, клієнти постійно вимагають, щоб бізнес-системи були достатньо адаптивними та реагували не лише на їхнє щоденне навантаження, але й надавали їм додаткові відомості, щоб вони могли приймати обізнані та більш надійні рішення [3].

По мірі того, як глобальна економіка змінюється у відповідь на розвиток нових технологій, бізнесу необхідно ставати більш гнучким та швидко реагувати на вимоги, цілі та виклики вимогливих клієнтів. Крім того, конкурентний та фінансовий тиск змушує компанії бути більш ефективними, заохочуючи їх шукати шляхи використання новітніх технологій та методологій, які могли б допомогти їм бути продуктивнішими за конкурентів, економити витрати та додавати вартості своїм бізнесам.

Автоматизація бізнес процесів (АБП) є високоінтелектуальним новаторським рішенням для впровадження цифрової трансформації у бізнес-процесах. АБП може покращити роботу людини, особливо у сфері обслуговування та процесів, що пов'язані з керуванням. Останнім часом через зростаючий тиск на покращення обслуговування та операційної ефективності АБП було впроваджено в сфері соціального обслуговування, фінансів, обліку, управління ланцюжками постачання та інформаційних технологій (ІТ).

Множинні дослідження розкрили багато переваг впровадження АБП в організації (наприклад, [2]). Однак АБП стає більш потужним, коли підтримується штучним інтелектом (ШІ) та когнітивною автоматизацією, створюючи нову технологію, а саме автоматизацію бізнес процесів з використанням штучного інтелекту (ІАБП).

Існують фундаментальні відмінності між АБП та ІАБП, такі як тип задачі, що ставиться. АБП стикається з рутинними завданнями, у свою чергу ІАБП стикається з завданнями, що пов'язані з оцінками вхідних даних та прийняттям рішень.

АБП слідує інструкціям, ІАБП приймає рішення та дає висновки. Сфера застосування АБП ширша. ІАБП має бути спрямованим на конкретну мету. Витрати та час, що необхідний, на впровадження ІАБП більші, ніж у випадку з АБП.

Отже впровадження використання штучного інтелекту в системи АБП робить бізнес-процеси більш інтелектуальними через повну їх автоматизацію без участі людини або за часткової участі людини.

АБП і ШІ зазвичай розглядаються як окремі підходи в науковій літературі [3]. Однак їх поєднання та поєднання інноваційних можливостей та трансформації можуть вирішити задачі, що виникають внаслідок об'ємних завдань, що потребують прийняття рішень, та покращити прийняття рішень в організаціях.

Згідно з оцінкою Forrester, до 2021 року понад чотири мільйони АБП будуть автоматизувати повторювані завдання, але фокус буде рухатися у бік інтеграції ШІ та вдосконалення аналітичних можливостей АБП. Так само Everest Group пояснила, що більшість покупців були задоволені рішеннями АБП, але АБП потребують аналітичних та когнітивних можливостей. Тому основна мета цієї роботи полягає в розгляді нової та новаторської технології, яка реалізує включення ШІ в автоматизований процес автоматизації.

Основна мета АБП – це імітування людської роботи, а потім заміна її АБП. Автоматизація бізнес процесів - це тип програмного забезпечення, яке імітує щоденні людські взаємодії з комп'ютерами. Це означає, що програмне забезпечення може замінити процеси, які раніше виконувалися людьми, такі як вхід у систему, введення даних, виконання робочих процесів та будь-яку дію на основі правил, виконану на комп'ютері, особливо взаємодії з CRM, такими як Salesforce або Hubspot, інформаційними системами, такими як SAP, електронною поштою та хмарними сервісами. АБП відтворюють людські дії без участі людини.

Це програмне забезпечення називається роботом, оскільки воно автоматично працює. АБП можуть читати електронні листи, відкривати вкладення, переміщувати файли, дотримуватися програмованих правил, видобувати дані, інтегрувати інформацію з ERP-системами, CRM-системами та системами управління людськими ресурсами, та багато іншого.

АБП має безліч сфер застосування: страхування, капіталовкладення, банківська справа, фінанси та облік, управління бізнесом та багато іншого. Цей тип автоматизації допомагає працівникам сконцентруватися на більш критичних завданнях, бути більш інноваційними та відводити час для підвищення своїх знань та навичок. Таким чином, технологія АБП демонструє зростання продуктивності та пропускну здатності при автоматизації повторюваних процесів.

Застосування АБП має певні особливості, які варто врахувати:

- АБП не потребують розробки абсолютно нової програмної інфраструктури, навпаки, вони передбачають, що ІТ-системи вже розроблені та працюють.
- Немає потреби заміни або зміни існуючих ІТ-систем, натомість потрібно інтегрувати керування АБП цими системами
- АБП має на меті створення програмного забезпечення, яке працює через інтерфейс користувача інших комп'ютерних систем так, як це робила би людина.

Але АБП має і свої недоліки та слабкі сторони, які відкривають двері до нових парадигм, методології таких як застосування ШІ. Деякі з таких недоліків:

- Необхідність детального та точного моделювання процесів автоматизації.
- Правила обробки вхідних та вихідних даних повинні бути чітко визначені відповідно до синхронізації між залежностями зовнішніх систем, що задіяні. АБП не вміє обробляти сценарії, які не були змодельовані раніше. Нові варіанти або сценарії вимагатимуть змін в логіці, правилах або введення даних в програмне забезпечення АБП.
- Великі за об'ємом даних процеси автоматизації потребують відповідної складної інфраструктури. АБП відмінно справляється з прийняттям рішень, але обмежена в складних процесах, які включають високій рівень абстракції.
- АБП не підходить для роботи з завданнями, пов'язаними з прийняттям рішень, і слід розглядати додавання штучного інтелекту.

Значна потреба у якісному попередньому аналізі та моделюванні бізнес процесів, неспроможність адаптації до умов, що змінюються, недостатня стійкість щодо великих обсягів даних, неоднорідність у прийнятті рішень і неможливість вирішення завдань, пов'язаних з прийняттям рішень, відкривають двері для впровадження ІАБП.

АБП можуть бути названі ІАБП, якщо вони обладнані можливостями штучного інтелекту, які допомагають подолати вищезазначені слабкості. Це досягається за допомогою вбудованих можливостей штучного інтелекту, таких як машинне навчання, глибоке навчання, обробка нативної мови, інші алгоритми та АРІ (інтерфейси програмування додатків), які допомагають у прогнозуванні, аналізі та оптимізації даних для вирішення проблем.

Агенція зі стандартизації IEEE визначає Інтелектуальну Автоматизацію Процесів (ІАБП) як "програмне забезпечення, що було налаштоване заздалегідь, яке поєднує бізнес-правила, логіку визначення контексту на основі досвіду та критерії прийняття рішень для ініціювання та виконання множини взаємопов'язаних людських та автоматизованих процесів у динамічному контексті" [4].

Інтелектуальна Автоматизація Процесів відрізняється від звичайної автоматизації своєю здатністю імітувати людські дії та відтворювати людське прийняття рішень на критичних етапах.

Наприклад, коли до традиційної автоматизації на основі правил додаються когнітивні можливості, виникає новий вид АБП. АБП реплікує раціональне прийняття рішень та інтелектуальні навички людини, використовуючи обчислювальні здібності для досягнення результатів, які перевищують людські можливості. ІАБП розширює функціональні можливості традиційного АБП за допомогою нових емерджентних технологій, таких як можливості самонавчання, виявлення процесів на основі вхідних даних, моделі оптимізації, розпізнавання екрану, алгоритми розпізнавання мовлення, розпізнавання зображень, машинне навчання та обробка нативної мови.

Популярність АБП зумовлена збільшенням швидкості та наявністю ресурсів для обробки процесів, що робить АБП більш високоефективним порівняно з людським робочим потенціалом. Наприклад, ці АБП є дешевшими, не потребують виплати за понаднормову роботу і працюють 24 години на добу і сім днів на тиждень. Крім того, вони забезпечують вищу точність, відповідність та швидкість.

Додаткова цінність АБП в основному пов'язана з покращенням роботи організації та зниженням витрат за рахунок збільшення працездатності за рахунок цифрових і недорогих АБП для виконання рутинних бізнес-процесів та переведення повноцінних працівників на виконання додаткових завдань, що поліпшує якість роботи. Поєднання АБП із ШІ дозволяє фундаментально збільшити сферу застосування АБП, а також вивести це на новий рівень. Застосування ШІ у сфері АБП є логічним та інноваційним рішенням.

Література

1. Asquith, A.; Horsman, G. Let the robots do it!—Taking a look at Robotic Process Automation and its potential application indigital forensics. *Forensic Sci. Int. Rep.* 2019, 1, 100007.
2. Leshob, A.; Bourgouin, A.; Renard, L. Towards a Process Analysis Approach to Adopt Robotic Process Automation. In *Proceedings of the 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), ICEBE 2018, Xi'an, China, 12–14 October 2018*;
3. Heinrich, B.; Klier, M.; Zimmermann, S. Automated planning of process models: Design of a novel approach to construct exclusive choices. *Decis. Support Syst.* 2015, 78, 1–14.
4. IEEE Std 2755-2017; IEEE Guide for Terms and Concepts in Intelligent Process Automation. IEEE: New York, NY, USA, 2017; pp. 1–16. Available online: <https://ieeexplore.ieee.org/document/8070671> (дата звернення 12 квітня 2024)

ВИКОРИСТАННЯ «TEXT-TO-SPEECH» МОДЕЛІ AUDIOAPI ВІД OPENAI

Зубко М.В., Лебединський А.В.

Харківський національний автомобільно-дорожній університет, Харків

У цій роботі ми розглянемо значення та можливості використання моделі Text-to-Speech (TTS) від OpenAI, відомої як AudioAPI, у сучасному інформаційному середовищі. «Text-to-Speech» є ключовою технологією, яка дозволяє автоматично перетворювати текст на мовлення людського стилю, що має великий потенціал у різних сферах життя.

Модель AudioAPI від OpenAI відрізняється високою якістю мовлення та інтонації, завдяки застосуванню глибоких нейронних мереж. Її застосування розглядається як в сфері технологій, так і в розважальній індустрії, що відкриває безліч можливостей для інновацій та покращення користувацького досвіду.

AudioAPI від OpenAI – це продукт, який дозволяє створювати реалістичне мовлення з використанням штучного інтелекту. Огляд моделі AudioAPI від OpenAI включає в себе детальне розглядання технологічних особливостей та потенціалу цієї системи.

Ця модель базується на передових методах глибокого навчання та нейронних мережах, що забезпечує високу якість мовлення. Вона натренована на великому обсязі аудіоданих, що дозволяє їй відтворювати мовлення з природним звучанням та інтонацією. Однією з ключових переваг моделі AudioAPI є її здатність генерувати реалістичне мовлення, яке легко розрізнити від синтезованого. Ця модель володіє широким набором голосових характеристик, що дозволяє користувачам вибрати підходящий для них стиль мовлення.

OpenAI's Text-To-Speech API автоматично розпізнає мову введення та генерує мову мовою введення тексту. API підтримує різні мови, зокрема англійську, китайську, гінді, іспанську, французьку, арабську, бенгальську, російську, португальську та індонезійську [2]. У той час як голоси за замовчуванням оптимізовано для англійської мови, API добре працює на інших мовах, адаптуючись до мови введення, щоб створювати розмовний звук потрібною мовою [3].

Користувачі повідомляють, що такі мови, як німецька, французька, іспанська та англійська, звучать майже ідеально без акцентів, демонструючи здатність API ефективно обробляти різні мови [2, 3]. Крім того, API дозволяє здійснювати потокове аудіо в реальному часі за допомогою кодування передачі фрагментів, що дозволяє відтворювати аудіо до того, як буде створено повний файл і стане доступним.

Функціональні можливості:

- гнучкість вибору голосів: AudioAPI надає можливість вибору різних голосів для генерації мовлення відповідно до ваших потреб;
- реалістичне мовлення: модель здатна створювати мовлення з природним інтонацією, ритмом та акцентом, надаючи вражаючу автентичність;
- підтримка різних мов: AudioAPI може працювати з різними мовами, що розширює її застосування на міжнародному рівні.

Застосування в технологіях:

- синтезатори мовлення для додатків: Модель AudioAPI може бути використана для створення синтезаторів мовлення у додатках [4], що дозволяє користувачам отримувати аудіо відтворення тексту, наприклад, для читання книг або повідомлень;
- асистенти для людей з обмеженими можливостями: Технологія TTS може бути корисною для людей з вадами зору або іншими обмеженнями, які затрудняють читання тексту. Модель AudioAPI може бути використана для створення асистентів, що допомагають їм отримувати аудіо відтворення текстової інформації.

Застосування в розвагах:

- аудіокниги та подкасти: використання моделі AudioAPI дозволяє створювати високоякісні аудіокниги та подкасти, що забезпечують природне мовлення та підвищують враження від аудіального контенту;
- відеоігри: технологія TTS може бути використана для надання голосових характеристик персонажам у відеоіграх, що покращує іммерсію гравців та забезпечує додатковий рівень взаємодії.

У висновку можна сказати, що використання моделі AudioAPI від OpenAI відкриває безліч можливостей у багатьох сферах, включаючи розваги, технології доступності та електронну комерцію. AudioAPI забезпечує можливість контролювати інтонацію мовлення, що робить її більш гнучкою для різних застосувань. Завдяки постійному вдосконаленню технологій штучного інтелекту, модель AudioAPI продовжує покращуватися в якості та швидкості генерації мовлення. Вона є потужним інструментом для розробників та дослідників, які працюють у сферах, пов'язаних з обробкою природної мови та аудіовиведенням. Її потенціал вже застосовується у різних галузях, від веб-розробки до поліпшення доступності контенту.

Список використаних джерел

1. Учасники проєктів Вікімедіа. Синтез мовлення – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Синтез_мовлення (дата звернення: 11.04.2024).
2. Can I choose the TTS language?. *OpenAI Developer Forum*. URL: <https://community.openai.com/t/can-i-choose-the-tts-language/477491> (date of access: 11.04.2024).
3. Text to Speech Guide. *OpenAI*. URL: <https://platform.openai.com/docs/guides/text-to-speech> (date of access: 11.04.2024).
4. Building With Bubble. How To Add OpenAI Text-To-Speech To Your Bubble App (Complete Guide), 2024. *YouTube*. URL: <https://www.youtube.com/watch?v=6dicvYNHyDw> (date of access: 11.04.2024).

ВИКОРИСТАННЯ ГЕНЕРАТИВНОЇ МОДЕЛІ ШТУЧНОГО ІНТЕЛЕКТУ DALL-E У ВЕБДОДАТКАХ

Кирилов Д.І., Лебединський А.В.

Харківський національний автомобільно-дорожній університет, Харків

У сучасному світі штучний інтелект (ШІ) використовується в різних сферах, зокрема, в обробці зображень. Одним із найновітніших розробок є генеративна модель DALL-E, створена OpenAI, яка здатна генерувати унікальні зображення на основі текстового опису. Ця модель, базуючись на трансформерній архітектурі [1, 2], здатна створювати зображення на основі текстового опису, відкриваючи безмежні можливості для творчості та інновацій. У даній доповіді розглянемо можливості використання моделі DALL-E у вебдодатках та її потенційні переваги.

Хоча DALL-E залишається одним з найбільш відомих генераторів зображень штучного інтелекту, на ринку з'явилося кілька альтернатив, які також користуються популярністю. Midjourney і Stable Diffusion – два з найвідоміших з них, порівняння яких зроблено у [3]. Midjourney, розроблений незалежною лабораторією в Сан-Франциско, знаходиться у відкритому бета-тестуванні та може бути використаний через Discord. Цей інструмент славиться високою якістю та деталізацією, проте потребує оплати за створення зображень. Stable Diffusion, який має відкритий вихідний код, спочатку навчався на 2,3 мільярдах зображень і розроблений командою дослідників з різних установ, включаючи CompVis Group, StabilityAI та RunwayML. Популярність Stable Diffusion зростає, а його функціональність постійно розвивається завдяки активній спільноті користувачів.

DALL-E має різноманітні застосування, наприклад, створення власних зображень для соціальних мереж, ігор і додатків віртуальної реальності. Його можна використовувати в різних сферах, включаючи рекламу, маркетинг, графічний дизайн, відеоігри, а також дизайн продуктів. Наступник DALL-E, DALL-E 2, призначений для створення більш фотореалістичних зображень із вищою роздільною здатністю [4]. Інтерфейс DALL-E зображений на рис. 1.

Модель DALL-E AI пропонує розробникам низку можливостей API для використання розширених можливостей створення зображень. Ось основні можливості API для моделі DALL-E:

Інтеграція у продукти:

- DALL-E API дозволяє розробникам інтегрувати найсучасніші можливості створення зображень безпосередньо у свої продукти, дозволяючи їм створювати різноманітні та високоякісні зображення з текстових описів [4];
- розробники можуть вибрати саме ті функціональні можливості, які їм потрібні, не навантажуючи свій вебдодаток непотрібними бібліотеками і додатковими функціями [4].

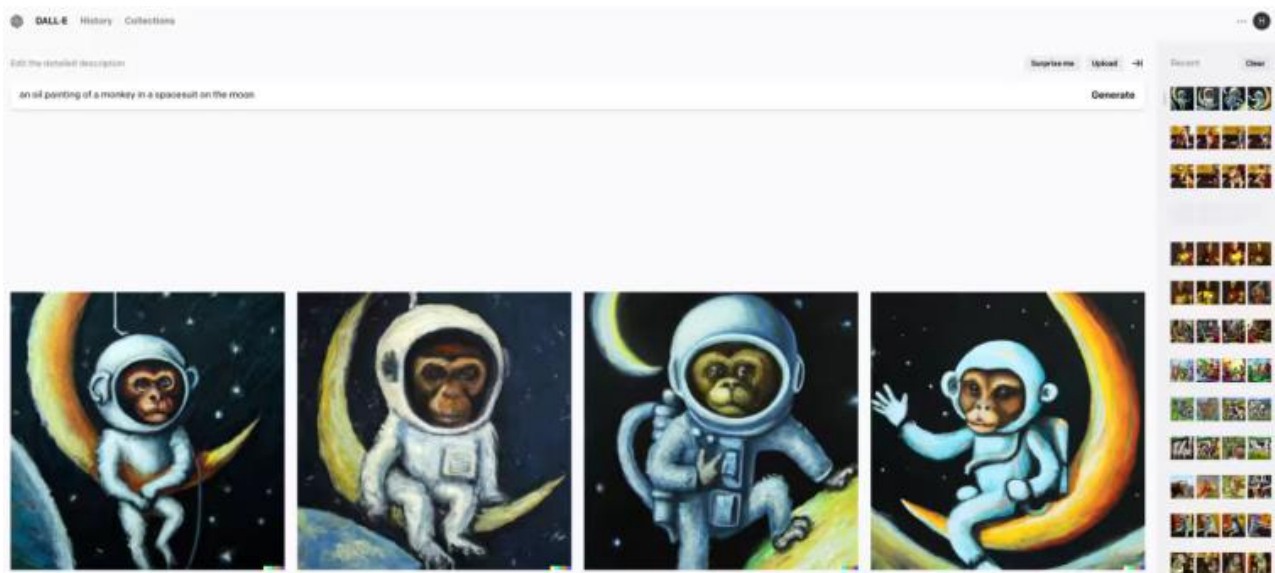


Рисунок 1 – Веб-інтерфейс моделі DALL-E

Модель «оплата по мірі використання»:

- DALL-E API працює на основі оплати за використання, що дозволяє користувачам отримувати доступ до його можливостей створення зображень на основі їх використання та потреб;

- інформацію про ціни на DALL-E API можна знайти на сторінці цін OpenAI, за запитом доступні варіанти знижок на великий обсяг.

Вихідне зображення та функції:

- DALL-E API може виводити зображення як URL-адреси або у форматі b64_json, забезпечуючи гнучкість доступу до згенерованих зображень і їх використання;

- за допомогою DALL-E 3 користувачі можуть скористатися такими функціями, як інтеграція тексту в зображення, що дозволяє бездоганно інтегрувати текст у візуальний контекст, а також можливість генерувати зображення в різних орієнтаціях, включаючи альбомний і портретний формати [5].

Саме наявність детальної документації для роботи з API дозволяє втілити у життя будь-які вебдодатки з використанням саме цієї функціональності. На рис. 2 показаний код функції generateImage, яка на вхід отримує аргументи вибору моделі, рядку запиту, роздільну здатність картинки, якість та кількість картинок.

Версії моделі та вдосконалення:

- API DALL-E підтримує різні версії моделі DALL-E, причому DALL-E 3 є останньою та найбільш вдосконаленою версією, яка пропонує покращене розуміння підказок і якість створених зображень [5];

- DALL-E 3 забезпечує зображення стандартної та високої чіткості з відповідною роздільною здатністю, задовольняючи потреби та вподобання різних користувачів [6].

```

1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.images.generate(
5     model="dall-e-3",
6     prompt="a white siamese cat",
7     size="1024x1024",
8     quality="standard",
9     n=1,
10 )
11
12 image_url = response.data[0].url

```

Рисунок 2 – Опис функції generateImage на NodeJS

Але з точки зору обмежень, створення зображень DALL-E не завжди може бути точним або відповідним, і можуть виникнути занепокоєння щодо порушення авторських прав, якщо створені зображення надто схожі на роботи, захищені авторським правом [7]. Модерація вмісту також є серйозною проблемою для запобігання створенню невідповідних, образливих або шкідливих зображень. Крім того, автоматизація створення контенту потенційно може витіснити робочі місця в таких сферах, як графічний дизайн, але також може відкрити нові ролі в нагляді та управлінні цими системами ШІ, але це вже інша тема для іншого дослідження.

Список використаних джерел

1. DALL-E 2. *OpenAI*. URL: <https://openai.com/dall-e-2> (date of access: 12.04.2024).
2. Учасники проєктів Вікімедіа. DALL-E – Вікіпедія. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/DALL-E> (дата звернення: 12.04.2024).
3. Midjourney vs. Stable Diffusion: Which Should You Use? - viso.ai. *viso.ai*. URL: <https://viso.ai/deep-learning/midjourney-stable-diffusion/> (date of access: 12.04.2024).
4. Demir D. A Guide to Using the DALL·E 3 API: How to Use and Test it Online. *Apidog Blog*. URL: <https://apidog.com/blog/dalle-3-api/> (date of access: 12.04.2024).
5. API Reference. Getting started. *OpenAI*. URL: <https://platform.openai.com/docs/api-reference> (date of access: 12.04.2024).
6. Keita Z. A Comprehensive Guide to the DALL-E 3 API. *Learn Data Science and AI Online | DataCamp*. URL: <https://www.datacamp.com/tutorial/a-comprehensive-guide-to-the-dall-e-3-api> (date of access: 12.04.2024).
7. Awan A. A. What is DALL-E?. *Learn Data Science and AI Online | DataCamp*. URL: <https://www.datacamp.com/blog/what-is-dall-e> (date of access: 12.04.2024).

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ У МЕДИЦИНІ

Кір'яков В.Д.

Харківський національний автомобільно-дорожній університет

На сучасному етапі інформаційної ери відбувається швидкий прогрес технологій та поширення програмних рішень, які спрямовані на покращення різних аспектів життя людини. Інформаційні технології швидко розвиваються в усіх сферах діяльності з кожним наступним роком. Активність людства стимулює необхідність пошуку та розробки нових методів використання нейронних мереж та їх можливостей для використання.

Нейромережі є нелінійними системами, які дозволяють краще проводити класифікацію даних, ніж традиційно застосовувати лінійні методи. Додаткові методи діагностики в медицині надають значно розширені можливості для проведення діагностики багатьох видів захворювань. Покращити специфічність методу, не покладаючи негативного впливу на його чутливість.

Гарною реалізацією багатьох ідей може стати саме машинне навчання (Machine Learning), глибоке навчання (Deep Learning) та комп'ютерний зір (Computer Vision). Ці методи можуть бути використані для вирішення задач різних форм складності в діагностиці. У сфері медицини можливо зосереджувати увагу саме на роботі з медичними приладами, які сильно пов'язані з процесом збору, обробки та аналізу даних щодо пацієнта. Тому різноманітні медичні зображення, такі як рентгенівські, комп'ютерні томографії (КТ) та цифрові гістологічні дослідження і т.д. сильно потребують високої точності у результативному висновку.

Одними із багатьох прикладів є:

1. Щодо ракових пухлин, у звітах Американського онкологічного товариства за 2012 рік було зафіксовано понад 1,6 мільйона випадків вперше діагностованого раку [1].

Отже, була поставлена задача розробити діагноз швидкого і адекватного клінічного ведення. Необхідні дані для діагностики було зібрано шляхом використання передових аналітичних методів, таких як мас-спектрометрія, і успішно використано в клінічній практиці для діагностики раку грудей і яєчників [2]. Штучні нейронні мережі також використовуються для діагностики різних типів пухлин головного мозку та раку легенів.

Алгоритм LYNA від Google AI може визначити наявність метастазів раку молочної залози з точністю практично 99% [2]. Такі ж самі алгоритми можуть виявляти переломи, крововиливи, ретинопатію, ураження шкіри, пневмонію, гіпертрофію серця, колапс легенів та аневризму. Для вирішення проблеми конфіденційності даних пацієнтів, дослідники компанії NVIDIA спільно з клінікою Майо та Центром дослідження клінічних даних розробили нейромережу, яка призначена для створення зображень головного мозку за допомогою

методу магнітно-резонансної томографії (МРТ). Ці зображення можуть бути використані для навчання алгоритмів діагностики.

Зокрема, при раку молочної залози вузлові метастази впливають на рішення про лікування щодо променевої терапії, хіміотерапії та можливого хірургічного видалення додаткових лімфатичних вузлів. Таким чином, точність і своєчасність виявлення вузлових метастазів має значний вплив на клінічну допомогу. Однак, дослідження показали, що приблизно 1 з 4 класифікацій стадії метастатичних лімфатичних вузлів буде змінено після другого патологічного огляду, а чутливість виявлення невеликих метастазів на окремому предметному склі може становити лише 38% при перегляді за часовими обмеженнями [2].

LYNA змогла точно визначити розташування ракових пухлин та інших підозрілих ділянок на кожному слайді, деякі з яких були занадто малі для виявлення їх патологоанатомами. В цілому, можна зробити висновок, що однією з потенційних переваг LYNA може бути виявлення цих проблемних областей для патологоанатомів, щоб вони могли розглянути й поставити остаточний діагноз.

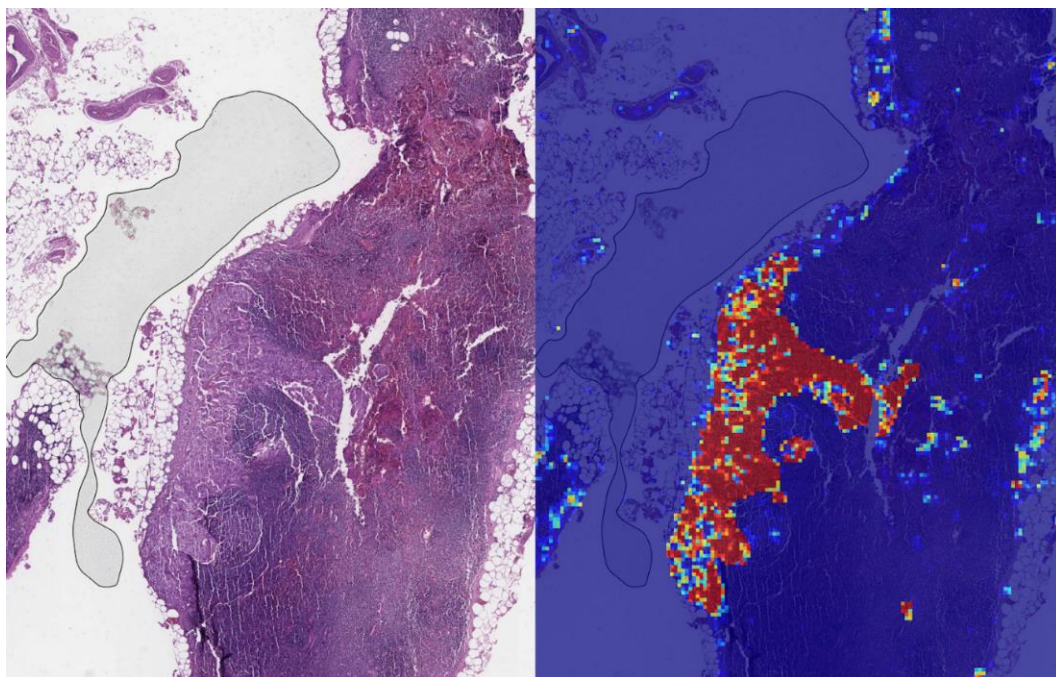


Рисунок 1 – Ліворуч: зразкове зображення, що містить лімфатичні вузли, з кількома артефактами: темна зона ліворуч – повітряний міхур, білі смуги – ріжучі артефакти, червоний відтінок у деяких областях – геморагічний (містить кров), тканина некротична (розкладається), а якість обробки була поганою.

Праворуч: LYNA визначає область пухлини в центрі

Ці дослідження мають суттєві обмеження, такі як обмежені розміри набору даних і змодельований діагностичний процес, який досліджував лише один слайд лімфатичних вузлів для кожного пацієнта замість кількох слайдів, які є звичайними для повного клінічного випадку. Для оцінки впливу LYNA на

реальні клінічні робочі процеси та результати пацієнтів необхідна подальша робота.

2. Новий інструмент аналізу ДНК, розроблений Університетом штату Мічиган, має здатність точно передбачати зріст людей і, що ще важливіше, потенційно оцінювати ризик розвитку серйозних захворювань, таких як хвороби серця і рак [3]. Цей інструмент або алгоритм створює прогнози для людських характеристик на основі генома, таких як зріст з точністю до 3 сантиметрів, щільність кісток. В основі цієї роботи можливо створити ще діагностику для таких хвороб як діабет, склероз та рак молочної залози.

У спадковому розумінні, спадковість відноситься до сукупності всіх генетичних впливів, включаючи нелінійні ефекти, такі як домінування або міжгенні взаємодії. Аддитивна спадковість відноситься до лінійних ефектів, які можна додавати: тобто передбачається, що кожен генетичний варіант має незалежний вплив на ознаку (який, звичайно, може бути рівним нулю), і всі вони додаються разом. Дані, які були проаналізовані, обмежені звичайними SNP (тобто одонуклеотидними варіантами, які зазвичай зустрічаються на рівні відсотків або більше в загальній популяції). Отже, були створені предиктори, які в найкращому випадку можуть виявити всю додаткову спадковість, обумовлену загальними SNP, для даної ознаки [3].

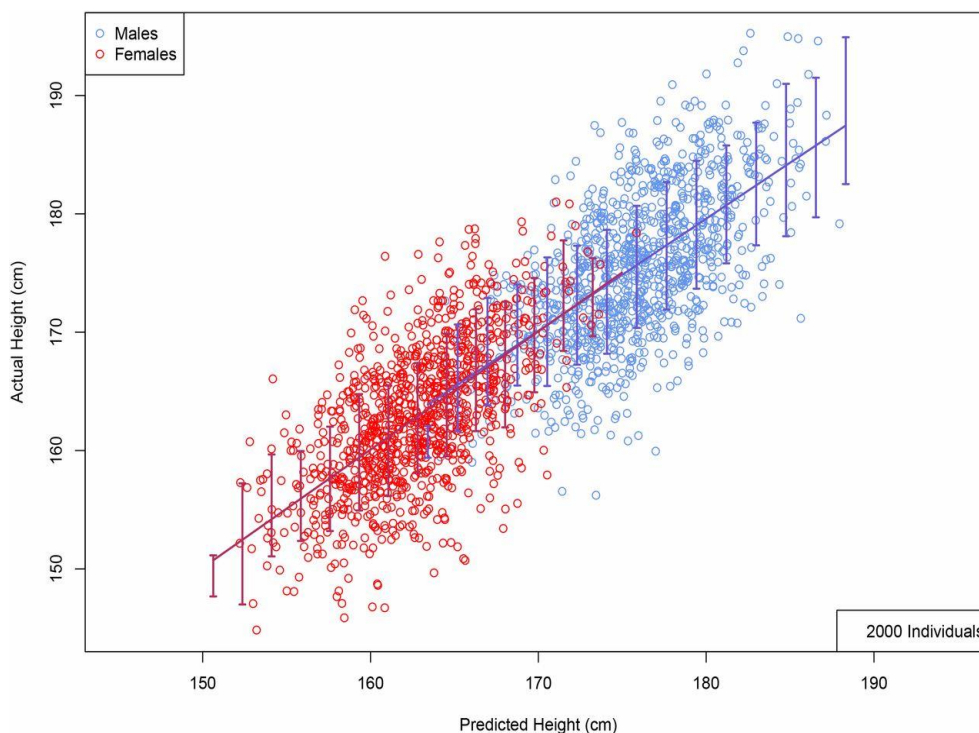


Рис 2. – Фактичний зріст у порівнянні з прогнозованим зростом з використанням 2000 випадково відібраних осіб, які не проходили навчання на предикторі.

3. Серцево-судинні захворювання є групою захворювань, які впливають на серце, його м'язи, судини, та вени. У своєму звіті Національний центр статистики Америки заявляє, що основним фактором смертності в США є серцево-

судинні захворювання [4]. Дані, отримані в минулому у результаті дали вплив для створення моделей нейронної мережі з використанням алгоритму зворотнього поширення помилок. За допомогою цієї моделі було досягнуто 80% точності в діагностиці цих захворювань на основі отриманих даних [5]. Та вони були використані для встановлення діагнозу конкретних типових серцевих захворювань

4. Діагностика діабету. Діабет утворив серйозну проблему для здоров'я. Кількість випадків діабету у світі оцінюється на рівні 366 мільйонів [6]. Діабет другого типу є типовим для цієї патології, що виникає внаслідок порушення клітинної реакції на інсулін, що приводить до гіперглікемії. Дані про вік, стать, вагу та рівень глюкози були зібрані та використані як вхідні параметри для розроблення штучної нейронної мережі, яка може представляти результати з точністю 90% [7]. Штучні нейронні мережі застосовуються для проведення оцінки рівня глюкози, а також для встановлення діагнозу діабету відповідно до біостатистичних аналізів клінічних досліджень.

Використання нейромереж у всіх процесах, пов'язаних з діагностикою в лікарні, дозволить досягти більшої точності та швидкості, що сприятиме зменшенню навантаження на лікарню та медичних працівників. Алгоритми з доступом до великих обсягів даних здатні прогнозувати виникнення або виявляти розвиток хвороби на ранній стадії, що дозволяє вчасно діагностувати та зупинити розвиток захворювання. Ці нові технології мають потенціал покращити якість та швидкість надання медичних послуг, роблячи їх більш зручними для як лікарів, так і пацієнтів. Проте, для досягнення цього необхідно вдосконалити роботу алгоритмів з метою мінімізації ймовірності помилки та вирішення проблеми безпеки конфіденційна інформація про пацієнтів. Вже сьогодні алгоритми можуть виконувати частину завдань з попереднього діагнозу на основі рентгенограм та знімків магнітно-резонансної томографії (МРТ).

Використовуючи схожі алгоритми у майбутньому, можливо створити комплексну систему, яка спостерігала б за людиною протягом усього її життя, мала доступ до її життєвих показників, отриманих за допомогою різних датчиків та сенсорів.

Список використаних джерел

1. Cancer Facts & Figures. American Cancer Society. National Home Office: American Cancer Society Inc. 2012, pp.9-24, [Internet]: <https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2012.html#:~:text=About%201%2C638%2C910%20new%20cancer%20cases>

2. Martin Stumpe, Technical Lead and Craig Mermel, Product Manager, Healthcare, Google AI Applying Deep Learning to Metastatic Breast Cancer Detection. 2018. [Online]: <https://blog.research.google/2018/10/applying-deep-learning-to-metastatic.html>

3. Lello L, Avery SG, Tellier L, Vazquez AI, de Los Campos G, Hsu SDH. Accurate Genomic Prediction of Human Height. Genetics. 2018 Oct 1;210(2):477–97. [Online]: <https://pubmed.ncbi.nlm.nih.gov/30150289/>

4. CDC. Heart disease facts. Centers for Disease Control and Prevention. 2023. . [Online]: <https://www.cdc.gov/heartdisease/facts.htm>

5. Kim JO (Ryan), Jeong YS, Kim JH, Lee JW. Machine Learning-Based Cardiovascular Disease Prediction Model: A Cohort Study on the Korean National Health Insurance Service Health Screening Database. *Diagnostics*. 2021 May 25;11(6):943. <https://doi.org/10.3390/diagnostics14020144>

6. Whiting DR, Guariguata L, Weil C, Shaw J. IDF Diabetes Atlas: Global estimates of the prevalence of diabetes for 2011 and 2030. *Diabetes Research and Clinical Practice*. 2011 Dec;94(3):311–21. doi: 10.1016/j.diabres.2011.10.029.

7. Ledisi G. Kabari: Diagnosing Diabetes Using Artificial Neural Networks. February 2020 *European Journal of Engineering and Technology Research* 5(2):221-224. DOI:10.24018/ejers.2020.5.2.1774

ГЛИБОКІ МОВНІ МОДЕЛІ: ВІД ТЕХНОЛОГІЧНОГО ПРОРИВУ ДО ПРАКТИЧНИХ ЗАСТОСУВАНЬ

Літвін Олександр, Пронін С.В

Харківський національний автомобільно-дорожній університет, Харків

Глибокі мовні моделі - це сукупність алгоритмів машинного навчання, які призначені для розуміння та генерації природної мови. Вони базуються на глибокому навчанні (deep learning) і використовуються для розв'язання різноманітних завдань у сфері обробки природної мови (NLP), таких як машинний переклад, синтез мови, розпізнавання мовлення тощо.

Огляд поняття глибоких мовних моделей

Глибокі мовні моделі відрізняються від традиційних методів обробки мови своєю здатністю адаптуватися та вчитися на великих обсягах даних. Вони використовують глибокі нейронні мережі з численними шарами (відси назва "глибокі") для автоматичного вивчення складних залежностей у тексті.

Історія розвитку в цьому напрямку

Історія розвитку глибоких мовних моделей пов'язана з виникненням та постійним вдосконаленням штучних нейронних мереж. Одним з ключових проривів стало введення архітектури Transformer, яка стала основою багатьох сучасних моделей. Починаючи з моделі BERT від Google до GPT (Generative Pre-trained Transformer) від OpenAI, глибокі мовні моделі стали домінуючими в галузі обробки природної мови.

Важливість глибоких мовних моделей у сучасному світі

Глибокі мовні моделі відіграють ключову роль у сучасному світі завдяки своїм можливостям в розумінні та генерації мови. Вони використовуються у багатьох галузях, від інтелектуальних асистентів до платформ штучного інтелекту. Їх важливість полягає в здатності автоматизувати та поліпшувати обробку природної мови, що відкриває нові можливості для розвитку технологій та покращення користувацького досвіду.

Технологічний прогрес у глибоких мовних моделях

Архітектура моделей: Transformer

Transformer - це архітектура глибоких нейронних мереж, яка вперше була запропонована у статті "Attention is All You Need" в 2017 році. Ця архітектура замінила традиційні рекурентні та зворотні нейронні мережі у задачах обробки природної мови. Основними складовими Transformer є механізм самоуваги (self-attention) та позиційне кодування, які дозволяють моделі ефективно опрацьовувати послідовності будь-якої довжини.

Підходи до навчання: Self-attention mechanism

Механізм самоуваги (self-attention) є ключовим елементом архітектури Transformer. Цей механізм дозволяє моделі приділяти різній увазі різним частинам вхідного тексту під час обробки. Він дозволяє моделі "фокусуватися" на важливих аспектах вхідного тексту, забезпечуючи кращу якість розуміння та генерації тексту.

Шляхи оптимізації та розвитку моделей

Шляхи оптимізації та розвитку глибоких мовних моделей включають в себе використання більших та якісніших наборів даних для навчання, оптимізацію алгоритмів навчання та пошук нових архітектур моделей. Також важливою є розробка методів передбачення та генерації тексту, які були більш точними та ефективними.

Використання глибоких мовних моделей у різних сферах

Обробка природної мови (Natural Language Processing, NLP)

Глибокі мовні моделі виявляються надзвичайно корисними в сфері обробки природної мови. Вони використовуються для розпізнавання та класифікації тексту, визначення настрою, екстракції інформації, сумаризації тексту та багатьох інших завдань. Завдяки глибоким мовним моделям можливо автоматизувати багато процесів, що пов'язані з обробкою тексту, що робить їх незамінними в багатьох сферах, від клінічної медицини до фінансів та маркетингу.

Машинний переклад

Глибокі мовні моделі перевершили традиційні методи машинного перекладу, забезпечуючи більш точні та природні переклади. Вони використовують контекстуальну інформацію та розуміння мови для генерації більш вірних перекладів, що робить їх незамінними для комунікації між мовами у різних галузях, від бізнесу до туризму.

Синтез мови

Глибокі мовні моделі також застосовуються у синтезі мови, тобто у створенні тексту на основі вхідних даних. Вони можуть бути використані для автоматичного створення описів продуктів, генерації новинних статей, писання креативного контенту та багатьох інших завдань, які вимагають генерації тексту.

Генерація тексту

Глибокі мовні моделі відіграють ключову роль у генерації тексту для різних цілей, від створення автоматичних відгуків у соціальних мережах до генерації творчого контенту у мистецтві та літературі. Вони можуть бути навчені на великих обсягах текстових даних та створювати текст, який максимально наближений до стилю та тематики вихідних даних.

Розпізнавання мови

Глибокі мовні моделі використовуються для розпізнавання мовлення у різних ситуаціях, від голосових асистентів на смартфонах до систем автоматичного підпису. Вони дозволяють точно розпізнавати інтонацію, акцент та інші аспекти мовлення, що робить їх корисними для різних завдань у галузі комунікації та інтерфейсів користувача.

Глибокі мовні моделі мають широкий спектр застосувань у різних сферах, від обробки природної мови до машинного перекладу та генерації тексту. Їх потужність та гнучкість роблять їх незамінними у сучасному світі комунікації та обробки інформації.

Виклики та перспективи

Обмеження сучасних моделей

Незважаючи на значний технологічний прогрес, сучасні глибокі мовні моделі мають свої обмеження. Одним з найбільших обмежень є нестабільність у роботі з малочисельними або неправильно підготовленими даними, що може призводити до неправильних результатів або викривлення виходу моделі. Також важливими є обмежені можливості мовних моделей у врахуванні контексту та семантики, що обмежує їхню точність у ряді завдань.

Етичні питання використання глибоких мовних моделей

Зростаюча потужність глибоких мовних моделей ставить перед нами ряд етичних питань. Наприклад, можливість створення фальшивих новин або маніпуляції громадською думкою за допомогою автоматичної генерації тексту. Також важливими є питання приватності та захисту персональних даних, оскільки глибокі мовні моделі можуть потенційно здійснювати аналіз та інтерпретацію особистої інформації.

Можливі напрямки подальшого розвитку

Незважаючи на виклики, перед глибокими мовними моделями відкриті широкі перспективи розвитку. Одним із напрямків є пошук нових архітектур моделей, що забезпечать кращу точність та гнучкість у різних завданнях. Також важливим є розробка методів адаптації моделей до конкретних доменів або мовних варіантів, що дозволить їх ефективніше використовувати у практичних застосуваннях. Подальший розвиток глибоких мовних моделей також пов'язаний з покращенням їхньої здатності до розуміння та генерації мови, що відкриває нові можливості для застосування в різних сферах, від освіти до медицини.

Хоча глибокі мовні моделі зустрічаються з викликами, вони також мають великий потенціал для подальшого розвитку та застосування в різних сферах життя. Розуміння цих викликів та перспектив може допомогти нам вдосконалити та використовувати ці моделі з максимальною користю для суспільства.

Приклади успішного застосування глибоких мовних моделей

Кращі практики у використанні глибоких мовних моделей

Модель BERT від Google відзначається використанням у класифікації тексту з вражаючою точністю, зокрема у сферах аналізу соціальних медіа та медичного тексту.

Netflix успішно використовує моделі на основі Transformer для рекомендацій фільмів та серіалів, що дозволяє надавати персоналізовані рекомендації користувачам.

Кейси успіху у відомих компаніях або дослідницьких проектах

1. **OpenAI і модель GPT:** Модель GPT від OpenAI використовується для створення природних текстових відповідей та новинних статей з вражаючою автономністю.

2. **Microsoft і проект Turing-NLG:** Проект Turing-NLG від Microsoft створює AI-систему, здатну відповідати на питання користувачів у реальному часі.

3. **Facebook та модель BART:** Модель BART від Facebook використовується для машинного перекладу, генерації тексту та сумаризації, покращуючи допоміжні інтерфейси та інтерактивні системи.

Джерела літератури

1. Васвані, А., Шазір, Н., Пармар, Н., та ін. (2017). "Увага - це все, що вам потрібно".
2. Девлін, Дж., Чанг, М. У., Лі, К., та Тоутанова, К. (2018). "BERT: Переднавчання глибоких двонапрямлених трансформаторів для розуміння мови".
3. Редфорд, А., Ву, Дж., Чайлд, Р., та ін. (2019). "Мовні моделі - це ненавчені багатозадачні вчені". Блог OpenAI, 1(8), 9.
4. Льюїс, М., Ліу, Ї., Гойяль, Н., та ін. (2020). "BART: Шумопідготовка передавання послідовностей-до-послідовності для генерації природних мов, перекладу та розуміння".
5. Браун, Т. Б., Манн, Б., Райдер, Н., та ін. (2020). "Мовні моделі - це навчені вчені".

МОДЕЛЬ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ТРАНСПОРТНОЮ СИСТЕМОЮ МІСТА

Мізяк І.О.

Харківський національний автомобільно-дорожній університет

Інтелектуально інформаційна система управління транспортною системою міста - це комплексний підхід до організації та управління транспортними потоками у місті за допомогою сучасних технологій штучного інтелекту.

Така система може поєднувати в собі різні компоненти, такі як сенсори, датчики, камери відеоспостереження, GPS-приймачі, різні способи передачі даних а також великі обчислювальні ресурси та алгоритми машинного навчання.

Основні функції моделі інтелектуальної інформаційної системи управління транспортною системою міста можуть включати:

1. Моніторинг і збір даних: система збирає дані з різних джерел, таких як сенсори на дорогах, камери відеоспостереження, мобільні пристрої тощо. Ці дані можуть включати інформацію про транспортні потоки, відстані, швидкості, густоту руху та інші параметри.

2. Аналіз даних: система використовує алгоритми машинного навчання та інші методи аналізу даних для виявлення транспортних проблем, прогнозування трафіку та визначення оптимальних шляхів управління транспортними потоками.

3. Управління сигналізацією: на основі аналізу даних система може автоматично адаптувати сигнальні режими на світлофорах та інших дорожніх знаках для оптимізації руху автомобілів.

4. Інформаційне сповіщення: система може надавати інформацію водіям та пасажиром щодо стану доріг, трафіку, оптимальних маршрутів та інших корисних даних через електронні табло, мобільні додатки або SMS-повідомлення.

5. Планування та оптимізація маршрутів: система може розробляти оптимальні маршрути для транспортних засобів, враховуючи поточний стан доріг і трафіку, що допомагає зменшити затори та забезпечити ефективніше використання дорожньої інфраструктури.

Модель інтелектуальної інформаційної системи управління транспортною системою міста може допомогти покращити рух транспорту, зменшити затори, зменшити рівень ДТП на дорогах загального користування, скоротити час простою та поліпшити загальну транспортну доступність міста.

У даній науково дослідній роботі буде розглянуто планування та оптимізація маршрутів для автомобілів спеціального призначення у зв'язку з великою кількістю дорожньо-транспортних пригод та небезпечних ситуацій на регульованих перехрестях, за участі автомобілів спеціального призначення.

Перехрестя є найнебезпечнішим місцем дорожньої мережі. Це пов'язано з тим Статистика за 2006 рік показує що 53% від загальної кількості ДТП, за рік, відбулися на перехрестях. Також перехрестя являють собою область макси-

мального впливу факторів взаємодії транспортного потоку з дорожніми умовами на виникнення ДТП. Для перехресть з відносно великою аварійністю застосовують метод рознесення транспортних потоків у часі, який реалізується за допомогою світлофорного регулювання. Цей метод до кінця не вирішує питання з підвищення безпеки руху, хоча знижує на 30% аварійність у порівнянні з іншими методами організації дорожнього руху.

Особливої уваги заслуговує проїзд перехрестя спецслужбами (служба порятунку, поліція, швидка допомога та інш.) на забороняючий сигнал світлофора, так як це особливо небезпечно.

Автомобілі спецслужб дуже часто по всьому світу потрапляють в аварії. В більшості випадків ці аварії відбуваються на перехрестях в момент коли автомобіль з усіма ввімкненими сигналами перетинає перехрестя на червоний.

На сьогодні проблематика проїзду перехрестя спецмашинами є актуальною. Можливий і безпечний варіант проїзду перехрестя без використання додаткових засобів, але на це потрібен час. На безпечний прибуття швидкою допомогою витрачається приблизно 10 хвилин загального часу їзди до хворого, приблизно такий же показник і в інших спецслужб. Швидкість реагування та прибуття на виклик грає важливу роль в суспільстві.

Проаналізувавши проблематику даного питання та відсутність існуючих систем для вирішення даної проблеми, прийнято рішення дослідити та знайти варіанти реалізації більш безпечного проїзду регульованого перехрестя автомобілями спецтранспорту.

Загальний опис математичної моделі

Загальний опис математичної моделі системи дистанційного управління світлофорами на перехрестях може різнитися залежно від конкретних вимог і характеристик системи.

1. Модель трафіку: Перш за все, необхідно мати модель трафіку, яка описує рух транспортних засобів на перехресті. Модель може включати інформацію про розмір і швидкість транспортних потоків, динаміку руху автомобілів, прогнозування зупинок, розгортання і інші параметри, які впливають на потік транспорту.

2. Стан світлофорів: Модель також повинна враховувати стан світлофорів на перехресті, включаючи поточний сигнал (червоний, жовтий, зелений) для кожного напрямку руху. Можна використовувати бінарну або категорійну змінну для представлення стану світлофорів.

3. Правила пріоритету: В моделі можна включити правила пріоритету, які визначають, коли спецтранспорт має отримувати пріоритетний проїзд. Дані правила можуть враховувати фактори, такі як тип і екстреність спецтранспорту, відстань до перехрестя, наявність інших транспортних засобів і т.д.

4. Алгоритм управління: В моделі повинен бути реалізований алгоритм управління, який враховує інформацію про трафік, стан світлофорів та правила пріоритету для прийняття рішень щодо переключення сигналів світлофорів.

Цей алгоритм може бути заснований на різних методах, таких як планування руху, оптимізація або прогностичні моделі.

5. Валідація та оптимізація: Математична модель може бути піддана валідації та оптимізації, щоб переконатися, що вона ефективно функціонує і задовольняє вимоги системи. Це може включати в себе аналіз результатів симуляцій, експериментів або використання оптимізаційних алгоритмів для знаходження оптимальних налаштувань системи управління.

Для розробки конкретної математичної моделі системи дистанційного управління світлофорами на перехрестях важливо мати детальні вхідні дані про перехрестя, характеристики трафіку, технічні можливості світлофорів та вимоги до системи. Також потрібна експертна думка і спеціалізовані знання з транспортного управління для розробки ефективної моделі.

Можливі варіанти реалізації

Для вирішення даної проблеми та покращення роботи спецслужб, необхідно створити комунікаційно-інформаційну систему з використанням бездротового зв'язку WiFi, який забезпечить передачу даних між необхідним автомобілем або центром управління та регульованим перехрестям.

Комунікаційно-інформаційна система управління призначена для забезпечення безпечного проїзду регульованого перехрестя. Суть системи полягає в автоматичному завчасному перемиканні сигналу світлофора в залежності від його поточного сигналу, в режимі реального часу. Цей комплекс функціонує на основі GPS та бездротової передачі даних WiFi. Принцип роботи системи показано на рис. 1.

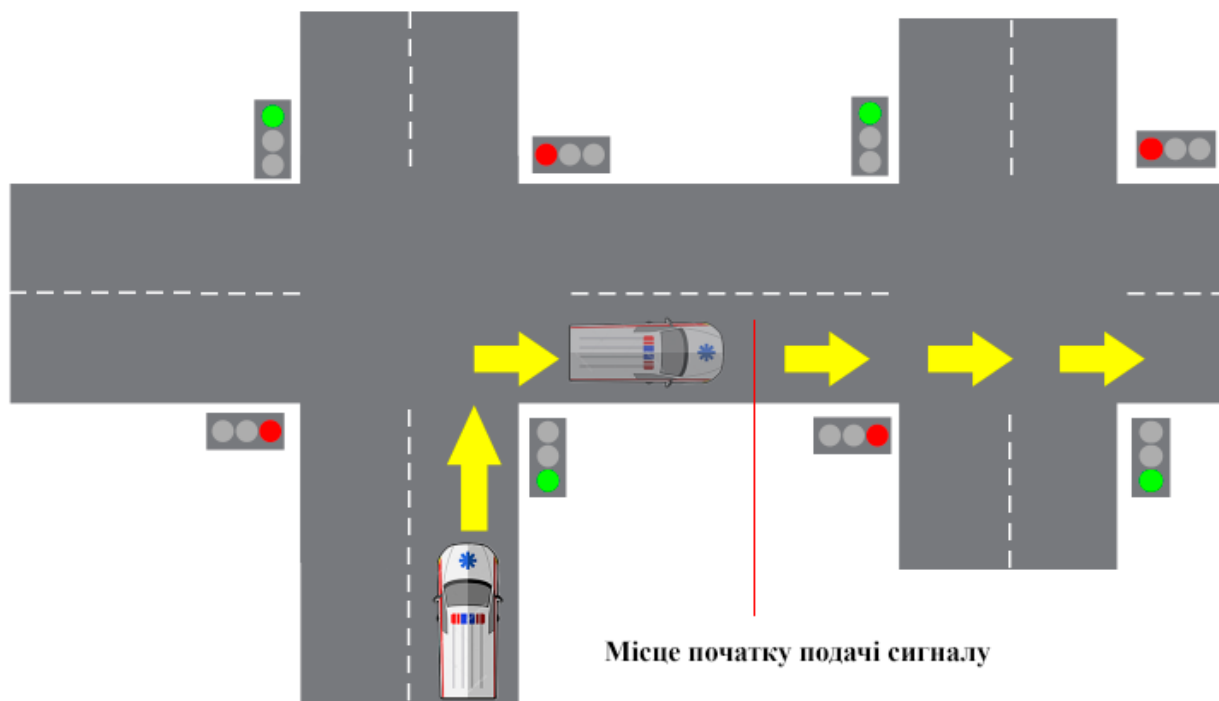


Рисунок 1 – Схема функціонування

Отримавши сигнал, світлофор відправляє сигнал підтвердження про успішне отримання даних та виконує необхідні дії в залежності від поточного сигналу світлофора. Система завершує свою роботу перемиканням на штатний режим роботи світлофора. Під час руху автомобіля на виклик, система аналізує маршрут на наявність регульованих перехресть, визначає координати автомобіля та проводить розрахунки для визначення моменту подачі сигналу.

Також потрібно враховувати що реалізація даного функціоналу вимагає належного проектування та інтеграції з існуючою системою управління світлофорами. Також потрібно враховувати місцеві правила та регуляції, які визначають допустимі способи і умови перемикання світлофора для спецтранспорту.

Перелік посилань

1. Great-circle distance [Електронний ресурс] / Режим доступу: https://en.wikipedia.org/wiki/Great-circle_distance
2. Національний стандарт України ДСТУ 4092-2002 «Світлофори дорожні. Загальні технічні вимоги, правила застосування та вимоги безпеки» [Електронний ресурс] / Режим доступу: http://www.infocar.com.ua/law_ukr/law_124.html
3. Transportation Engineering and Planning by C.S. Papacostas, P.D. Prevedouros
4. Handbook of Transportation Engineering by Myer Kutz
5. Control and Automation of Electrical Power Distribution Systems by James Northcote-Green, Robert G. Wilson

ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ У РОЗРОБЦІ ДИЗАЙНУ САЙТІВ

Решитько В.С., Карпішен Б.С.

Харківський національний автомобільно-дорожній університет, Харків

У двадцять першому столітті неможливо представити людину без інтернету та технологій пов'язаних з ним. Одним із найважливіших інструментів для взаємодії людини з інтернетом є сайти. У наш час буквально все можна зробити через інтернет: замовити їжу, придбати будь-яку річ, навчатися, працювати, розважатися, тощо. Це дозволяє економити купу часу, сил та нервів, що особливо актуально у такий важкий для України період. Саме ці фактори роблять професію WEB-розробника привабливою для освоєння.

Незважаючи на всі переваги професій пов'язаних з WEB-розробкою, багатьох новачків відлякує той величезний кейс навичок які повинен мати розробник, бо успішний фахівець повинен вміти розробити гарний дизайн, макет, логіку сайту, а також подбати про сумісність з різними платформами та браузерами. Але галузь не стоїть на місці та вже зараз розроблено достатньо інструментів, щоб поліпшити життя розробника. Саме одним із таких інструментів, що можуть значно спростити життя, є нейромережі. У WEB-розробці, а

саме під час створення дизайну нейронні мережі вже використовуються за наступними основними напрямками:

Аналіз поведінки та уподобань користувача. Єдиним інструментом, що може провести аналіз потреб користувача на основі його поведінки та уподобань з точністю наближеної до ідеальної є штучний інтелект. Правильно розроблений алгоритм може визначати платформи та дизайни, які подобаються різним користувачам. В свою чергу правильно розроблений дизайн з урахуванням порад нейромережі може охопити максимально велику аудиторію та значно збільшити ефективність сервісу. Також нейронна мережа може аналізувати контент сайту та пропонувати варіанти його оптимізації для пошукових систем. Це включає оптимізацію тегів, заголовків і основного тексту під конкретні ключові слова та пошукові запити з ціллю зробити контент максимально актуальним та зручним для пошуку.

Генерування елементів дизайну під час розробки. Головні якості, які поки недоступні штучному інтелекту – нестандартне мислення та креативність. Нейронна мережа не може повністю замінити дизайнера та розробити замість нього дизайн, тому що для більшості таке завдання занадто складне, а якщо і посильне, то такий сайт наврядче буде коректним та ефективним. Але нейромережа може стати незамінним помічником для створення окремих елементів дизайну та надання їм унікальних стилів.

Таким чином штучний інтелект може створювати ілюстрації, логотипи, фавікони чи елементи дизайну сайту. Розглянемо одного з представників мереж, що можуть генерувати зображення за запитом, а саме MidJourney [1]. Мережа взаємодіє з користувачем за допомогою чат-боту, що приймає текстовий опис зображення. Далі мережа генерує зображення та надає можливість редагування деталей, стилю чи якості результату. За допомоги цієї мережі можна створювати навіть ілюстровані книги. Наприклад, на прохання згенерувати зображення короля лева інтелект може видати результат зображений на рис. 1.



Рисунок 1 – результат роботи MidJourney

Розробка унікальних логотипів. Раніше, щоб створити унікальний логотип треба було вміти малювати або засвоїти ряд навичок зі створення графічних малюнків та редагування зображень у фото редакторі. Крім того на фрілансі існував навіть окремий вид діяльності пов'язаний зі створенням логотипів на замовлення. Зараз же достатньо зробити запит спеціально призначеній нейронній мережі і вона видасть вам унікальний логотип, краще якого не зможе зробити більшість графічних дизайнерів та художників. З одного боку поява подібних інструментів ускладнила життя вузькоспеціалізованим фахівцям але значно полегшила роботу WEB-розробникам. Гарним прикладом мережі, що спеціалізується на створенні логотипів є Logolivery [2]. Цей штучний інтелект дозволяє шляхом введення параметрів у чат-бот створювати високодеталізовані SVG логотипи з подальшою можливістю коригування результату. Генеруючи один сет логотипів можна отримати різні концепції одного зображення у багатьох можливих стилях. Після створення можна вручну відредагувати новостворений логотип та адаптувати відповідно до свого бренду. Крім того умови використання мережі надають повні права на комерційне використання логотипів та інтегрування лого у будь-які маркетингові матеріали, веб-сайти та продукти. Приклад роботи мережі наведений на рис. 2.

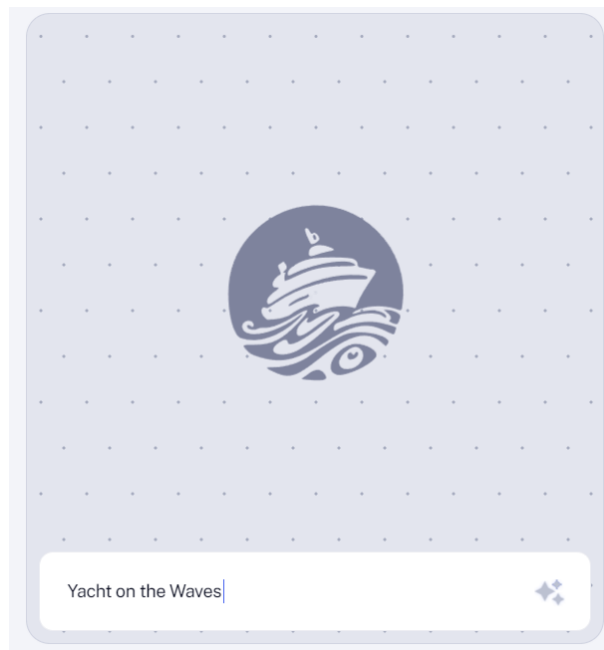


Рисунок 2 – Приклад роботи нейронної мережі Logolivery

Здатність нейронних мереж замінювати людей. Завдяки гігантським темпам розвитку технологій штучного інтелекту та зростаючим попитом на них, усіх фахівців, роботу яких можуть виконувати нейронні мережі, законо-мірно почала турбувати проблема можливості їх заміни. Алгоритми роботи штучних інтелектів, ще не досягли достатнього рівня розвитку, щоб надати конкуренцію середньостатистичному фахівцю. Крім того усі нейронні мережі в основному навчаються на загальнодоступних базах, інформація у яких не

завжди достовірна та інколи обмежена певним періодом, що може призвести до неправильної відповіді. Наприклад, ChatGPT обмежений джерелами інформації періодом до кінця 2022 року, про що сама мережа повідомляє на початку роботи [3]. Також, тільки людині притаманні такі риси як нестандартне мислення та креативність. Усі перелічені фактори підтверджують той факт, що штучному інтелекту, ще досить довго прийдеться еволюціонувати для того, щоб розпочати заміну навіть простих професій.

Внутрішнє влаштування нейромереж. DALL-E – це штучний інтелект створений компанією OpenAI та призначений для роботи із зображеннями. DALL-E являє собою багатомодальну реалізацію технології GPT-3, що має дванадцять мільярдів параметрів для генерування відповідей [4]. Для того, щоб більш точно розуміти принципи роботи нейронних мереж розберімо внутрішнє влаштування штучного інтелекту ChatGPT.

Generative Pre-trained Transformer або ChatGPT, було розроблено з використанням методів керованого машинного навчання і машинного навчання з підкріпленням. Обидва способи представляють із себе тренування мережі за допомогою людей-асистентів [5].

Кероване машинне навчання (Supervised Learning) представляє собою набір методик, що ґрунтуються на технологіях машинного навчання, які дають змогу виявити функціональні взаємозв'язки в даних, що підлягають аналізу [6].

При керованому навчанні або навчанні з учителем користувачеві надається набір даних, який складається з вхідного значення x та відповідного вихідного y . Ціль полягає в тому, щоб побудувати класифікатор або регресор, який може оцінити вихідне значення для раніше невидимих входів [7].

Спочатку нейромережу натреновують стандартними методами навчання на вже існуючих базах даних. Потім модель тренується на діалогах, де людина виконує дві ролі: користувача і асистента штучного інтелекту. На етапі підкріплення тренер оцінює базу відповідей, що давала мережа у попередніх тестових діалогах. Ці оцінки використовуються для створення нагородних моделей, на яких інтелект доопрацьовується шляхом проходження багатьох ітерацій за принципом Proximal Policy Optimization [8].

Кероване машинне навчання. Задано набір з N -тренувальних даних вигляду:

$$\{(x_1, y_1), \dots, (x_N, y_N)\} \quad (1)$$

де x_i - вектор ознак і – ого прикладу даних, а y_i – мітка прикладу.

Для цього набору даних алгоритм шукає функцію вигляду:

$$g: X \rightarrow Y \quad (2)$$

де X – вхід, а Y – вихід, g – функція або елемент деякого простору можливих функцій так званого простору гіпотез.

Хоча просторові та оцінкові функції можуть приймати вигляд будь-якої функції з простору, багато навчальних алгоритмів являють з себе саме

ймовірнісні моделі, де просторова функція набуває вигляду умовної ймовірності:

$$g(x) = P(y|x) \quad (3)$$

Оцінкова функція набуває в свою чергу вигляду:

$$f(x, y) = P(x, y) \quad (4)$$

Для вибору функцій даних зазвичай використовують один з основних методів:

- Метод мінімізації емпіричного ризику;
- Метод мінімізації структурного ризику.

Мінімізацією емпіричного ризику шукають функцію, яка найкраще підходить до тренувальних даних. У Мінімізації структурного ризику створюють штрафну функцію, яка контролює компенсацію дисперсії.

Навчання з підкріпленням. Це галузь машинного навчання, що досліджує дії, які повинні виконувати програмні агенти в певному середовищі задля максимізації конкретного для певної ситуації сукупної винагороди.

Загальна модель методу виглядає наступним чином:

- Визначення множини станів середовища S ;
- Визначення множини дій A ;
- Визначення правил переходу між станами;
- Визначення правил визначення скалярної безпосередньої винагороди;
- Визначення правил опису спостережуваних агентом даних.

Одним з найпростіших методів дослідження опису даних є метод ϵ -жадбна стратегія. Агент вибирає дію, яка за його переконанням має найкращий довготерміновий ефект, з імовірністю $1 - \epsilon$, а інші дії обирає рівномірно стохастично. Далі відбувається корегування правил згідно вибору агента.

Висновки. У наш час штучний інтелект використовується у багатьох аспектах людського життя і почав ставати невід'ємною частиною нашого світу. Він може допомагати у навчанні, генерувати інформаційні статті, займатися маркетингом, оптимізувати робочі процеси, керувати базами даних, аналізувати інформацію, бути гарним опонентом у комп'ютерній грі, створювати відео чи зображення та виконувати безліч інших процесів.

Для WEB-розробника гарно підібрана нейронна мережа може стати помічником у створенні окремих елементів дизайну сайту, генеруванні зображень, наповненні сторінок інформацією, аналізі вподобань користувачів чи маркетингу.

Але, незважаючи на всі переваги використання нейронного помічника, варто не забувати, що нейромережі не можуть цілком замінити фахівця, що зумовлено недостатнім розвитком алгоритмів роботи, наявністю штучних обмежень розвитку технології з боку людини, обмеженою базою для навчання та відсутністю притаманних людині креативності і нестандартного мислення.

Перелік посилань

1. Midjourney [Електронний ресурс]. – Режим доступу: https://www.midjourney.com/home_
2. Logolivery [Електронний ресурс]. – Режим доступу: https://logolivery.ai_
3. ChatGPT [Електронний ресурс]. – Режим доступу: <https://chat.openai.com>.
4. Tamkin Alex; Brundage Miles; Clark Jack; Ganguli Deep. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models. 2021, 8 p.
5. Knox, W. Bradley; Stone, Peter. Augmenting Reinforcement Learning with Human Feedback. University of Texas at Austin, Department of Computer Science, 2011, 8 p.
6. О. М. Верес, Р. М. Оливко. Класифікація методів аналізу Великих даних. Вісник Національного університету "Львівська політехніка". Серія : Інформаційні системи та мережі. 2017, № 872, С.84-92.
7. Синєглазов В., Чумаченко О. Укладач Бідюк, П. І., Шугалей, Л. П. Методи та технології напівкеруваного навчання: Курс лекцій. Київ, КПІ ім. Ігоря Сікорського, 2022. С. 7-9.
8. OpenAI. ChatGPT: Optimizing Language Models for Dialogue. – 2022. – Режим доступу: <https://openai.com/blog/chatgpt>.

НЕЙРОННА МЕРЕЖА ДЛЯ ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ

Соколовський А.О.

Харківський національний автомобільно-дорожній університет

Прогнозування значень часового ряду за допомогою моделі MLP є складним завданням, що вимагає систематичного та методичного підходу. Починаючи з розуміння суті задачі, важливо визначити, яка саме змінна має бути прогнозована та які фактори впливають на цю змінну в часі.

Після аналізу часового ряду та виявлення його особливостей, таких як тренди, сезонність та аномалії, наступним етапом є підготовка даних. Це може включати в себе видалення шуму, виявлення та корекцію викидів, а також розгляд можливості стандартизації або нормалізації даних.

Будівництво моделі MLP передбачає визначення її архітектури, включаючи кількість шарів, кількість нейронів у кожному шарі та вибір функцій активації.

Навчання моделі включає в себе вибір оптимального алгоритму оптимізації, розділення даних на тренувальний та валідаційний набори, а також моніторинг функції втрати під час тренування. Ефективна оцінка моделі базується на використанні відповідних метрик, таких як середньоквадратична помилка (MSE) чи середньоабсолютна помилка (MAE), та аналізі залишкових графіків.

Етап гіперпараметризації та налаштування дозволяє експериментувати з параметрами моделі, такими як швидкість навчання чи кількість епох, для досягнення оптимальної продуктивності. Важливо також використовувати

крос-валідацію для підтвердження стійкості та генералізації моделі на різних наборах даних.

Після успішного навчання модель готова до прогнозування на нових даних та впровадження в робоче середовище. Систематичний моніторинг її продуктивності та можливе оновлення відповідно до змін в умовах завдання чи в самому часовому ряду є ключовим етапом в подальшому використанні моделі.

Характеристика об'єкта дослідження

Об'єктом дослідження є датасет "Air Passenger", який охоплює історичні дані щодо пасажирських перевезень авіатранспортом. Ці відомості включають кількість пасажирів, які скористалися авіаперевезеннями протягом різних періодів часу. За допомогою цього датасету прагнемо розробити прогностичну модель, базуючись на нейронній мережі з архітектурою MLP.

Однією з ключових складових датасету є інформація про дати проведення перевезень та відповідні обсяги пасажиропотоку. Ці дані формують часовий ряд, що репрезентує динаміку змін у кількості пасажирів з плином часу. Проведення аналізу цього часового ряду є невід'ємною частиною для розуміння його структури та виявлення можливих закономірностей, таких як сезонні коливання, тренди або випадкові величини.

Головною метою є створення прогностичної моделі, яка здатна точно прогнозувати майбутні значення пасажиропотоку. Використання нейронної мережі з моделлю MLP надає можливість адаптуватися до складних закономірностей в даних та забезпечує ефективне прогнозування.

Отримані результати дослідження можуть стати важливим внеском у розуміння та оптимізацію пасажирських перевезень в авіаційній галузі. Поєднання аналізу часового ряду та застосування нейронної мережі відкриває можливості для розробки ефективних стратегій управління ресурсами та планування на основі точних прогнозів.

Реалізація створеної нейронної мережі

Для реалізації нейронної мережі в коді використовується бібліотека `scikit-learn`, а саме клас `MLPRegressor`, який надає можливість створювати та навчати штучні нейронні мережі для задач регресії. Відома також як мультишаровий перцептрон (MLP), ця модель є однією з популярних архітектур нейронних мереж. Процедура тренування штучної нейронної мережі включає ініціалізацію моделі з визначенням її архітектури, навчання моделі на тренувальних даних та використання навченої моделі для прогнозування на нових даних.

Практична реалізація відбувається завдяки використанню бібліотек та інструментів мови програмування Python, зокрема бібліотеки `scikit-learn` для роботи з машинним навчанням та нейронними мережами. Результати виводяться на графіку з використанням бібліотеки `matplotlib`.

Код реалізації розбитий на кілька етапів для зручності розробки та розуміння процесу. Основні етапи включають:

1. Завантаження та Підготовка Даних: в цьому етапі відбувається завантаження даних з CSV-файлу за допомогою бібліотеки `pandas`. Конвертація стовбця 'Month' у формат дати та встановлення його як індекс. Розділення даних на тренувальний та тестовий набори.

2. Масштабування даних: на цьому етапі використовується `MinMaxScaler` для масштабування значень у діапазон від 0 до 1.

3. Створення часових послідовностей: відбувається визначення функції `create_sequences` для створення часових послідовностей для навчання та тестування.

4. Навчання моделі MLP: Визначення та навчання моделі `MLPRegressor` з використанням даних тренувального набору.

5. Прогнозування на тестових даних: використання навченої моделі для прогнозу значень на тестовому наборі.

6. Оцінка якості моделі: розрахунок середньоквадратичної помилки (MSE) для оцінки точності прогнозів.

7. Прогнозування майбутніх даних: прогноз майбутніх значень, використовуючи останні точки тестового набору.

8. Зворотне масштабування: зворотне масштабування прогнозованих та фактичних значень для відображення на графіку.

9. Візуалізація результатів: використання `matplotlib` для створення графіка, на якому відображаються фактичні дані, прогнозовані значення на тестовому наборі та майбутні прогнози.



Рисунок 1 – Результат роботи програми

Під час виконання була використана штучна нейронна мережа (MLP), яка є видом глибокого навчання, для вирішення задачі прогнозування пасажиро-

потокі на основі часового ряду пасажирських перевезень. Починаючи з завантаження та обробки даних за допомогою бібліотеки Pandas, дати у стовпці "Month" конвертуються у формат дати. Далі дані масштабуються до діапазону від 0 до 1 за допомогою MinMaxScaler з бібліотеки scikit-learn, що дозволяє нормалізувати дані для ефективного тренування моделі.

Цей код не лише демонструє процес прогнозування пасажиропотоку, але і надає основу для розвитку та удосконалення моделей прогнозування часових рядів з використанням нейронних мереж.

Висновок

У рамках виконаної роботи з прогнозування часового ряду пасажиропотоку використано штучну нейронну мережу (MLP). Перший етап включав підготовку та завантаження даних, де часовий ряд був оброблений та встановлено у формат datetime. Далі, для забезпечення стабільності навчання, дані були нормалізовані за допомогою MinMaxScaler.

Слідом за цим була реалізована функція для створення часових послідовностей, що допомагала враховувати залежності в часовому ряді. Нейронна мережа (MLPRegressor) була визначена та навчена на підготовлених даних, з врахуванням визначених параметрів архітектури та оптимізації.

Застосовано навчену модель для прогнозування значень на тестових даних, а якість прогнозу оцінено за допомогою середньоквадратичної помилки (MSE). Крім того, використано модель для прогнозування майбутніх значень часового ряду. Остаточні результати були візуалізовані за допомогою Matplotlib, де порівнювалися фактичні, прогнозовані та майбутні значення. Всі ці етапи взаємодіють, допомагаючи створити ефективний інструмент для прогнозування та визначення тенденцій в часовому ряді пасажиропотоку.

Отже, використання MLP дозволило створити модель, яка ефективно прогнозує значення часового ряду пасажиропотоку. Підготовка даних, правильний вибір архітектури та параметрів моделі, а також оцінка її точності через візуалізацію допомогли створити функціональний інструмент для прогнозування майбутніх значень в даному контексті.

Список джерел

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50 – P. 159–175.

Python Documentation - Scikit-learn. .: [Електронний ресурс].-Режим доступу: <https://scikit-learn.org/stable/documentation.html>

Pandas Documentation. .: [Електронний ресурс].-Режим доступу: <https://pandas.pydata.org/pandas-docs/stable/index.html>

Каллан, Р. Нейронні мережі: Короткий довідник / Р. Каллан. - М .: Вільямс І.Д., 2017. - 288 с.

Paolo Galeone Dropout — Методи рішення проблеми перенавчання в нейронних мереж [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/company/wunderfund/blog/330814/>.

J. Park, J. I.W. Sandberg. Universal approximation using radial-basis-function networks. *Neural Computation*, 1991. – 151 с.

НЕЙРОННА МЕРЕЖА ДЛЯ ПРОГНОЗУВАННЯ ТОНАЛЬНОСТІ ТЕКСТУ

Товстолуг Д. В.

Харківський національний автомобільно-дорожній університет

У сучасному інформаційному суспільстві обробка та аналіз текстової інформації стає все більш важливою задачею. Однією з ключових областей застосування є розпізнавання тональності тексту, що дозволяє визначити емоційний відтінок висловлювання. Це має велике значення в сфері аналізу відгуків, моніторингу громадської думки, та рекомендаційних систем.

Глибокі нейронні мережі, зокрема мережі з рекурентними шарами, стали ефективним інструментом для вирішення завдань аналізу тексту. У цьому контексті, GRU (Gated Recurrent Unit) представляє собою вдосконалену архітектуру рекурентної мережі, яка володіє здатністю враховувати контекст і зберігати довгострокові залежності в тексті.

Наше дослідження спрямоване на використання GRU мережі для розпізнавання тональності тексту. GRU – це вдосконалена рекурентна архітектура, яка дозволяє враховувати контекст та зберігати довгострокові залежності в тексті.

Для нашого дослідження ми використовуємо Stanford Sentiment Treebank (SST) датасет – збірку текстів, які включають анотації з тональності в межах від 0 до 1. Цей датасет надає різноманітність та деталізацію у вираженні емоційного відтінку, що важливо для нашого дослідження розпізнавання тональності тексту за допомогою GRU мережі.

Характеристика об'єкта дослідження

Датасет, що використовується для розпізнавання тональності тексту, є поєднанням двох файлів із Stanford Sentiment Treebank датасету. Перший файл (sentlex_exp12.txt) містить інформацію про тексти за їхніми ідентифікаторами, а другий файл (sentiment_labels.txt) містить ідентифікатори текстів і відповідні їм значення тональності у вигляді чисел в діапазоні від 0 до 1. Тексти та їхні тональності об'єднуються за ідентифікаторами.

Stanford Sentiment Treebank містить тексти з різних джерел, що дозволяє створити модель, здатну розпізнавати тональність для широкого спектру мовленнєвих виразів.

Датасет був створений колективом науковців та лінгвістів для сприяння дослідженню та розвитку методів аналізу тональності та обробки природної мови.

Sentiment Treebank є корпусом текстів, розмічених щодо тональності, що був створений для використання в розробці та валідації моделей машинного навчання в галузі аналізу тональності тексту. Датасет містить рецензії на фільми з відгуками, що надають контекст реального вживання мовлення.

Датасет містить тексти рецензій, які розбиті на фрази та речення, і кожному з цих фрагментів призначено значення тональності у вигляді числових оцінок в діапазоні від 0 до 1, де 0 відповідає негативній тональності, 1 – позитивній, а проміжні значення відображають неоднозначність чи нейтральність.

Одним з унікальних аспектів STT є наявність анотацій та деревинної структури, що дозволяє моделям розуміти взаємозв'язки між частинами тексту та їхні впливи на загальний сентимент.

STT є важливим ресурсом для розробників та дослідників, оскільки він надає багатий і різноманітний набір даних для створення та вдосконалення моделей аналізу тональності.

Аналіз предметної галузі

Розпізнавання тональності тексту відноситься до задач аналізу тексту і природної мови.

Аналіз тексту та природної мови (NLP) відноситься до області комп'ютерної науки, що спеціалізується на розумінні та обробці людської мови комп'ютерами. Ця галузь вивчає методи та технології, спрямовані на взаємодію між комп'ютерами та природною мовою, що дозволяє системам аналізувати, розуміти та взаємодіяти з текстовою інформацією.

Завдяки розвитку методів машинного навчання та глибокого навчання, системи NLP стають все більш вдосконаленими в аналізі текстової інформації. Вони використовуються для створення чат-ботів, автоматизації відповідей на електронні листи та для покращення пошукових систем.

Вивчення тональності тексту є важливим в сучасному суспільстві, особливо в галузі бізнесу. Підприємства використовують аналіз тональності для визначення реакції споживачів на їхні продукти та послуги, виявлення проблем та взаємодії з аудиторією в соціальних мережах.

NLP використовується для створення систем, які можуть розпізнавати та реагувати на емоційний стан користувачів. Це актуально для розваг, освіти та інших сфер.

В області медицини NLP використовується для аналізу медичної літератури, електронних медичних записів та взаємодії з пацієнтами через інтелектуальні системи.

Аналіз текстових даних застосовується в фінансовій сфері для визначення трендів, прогнозування ринкових змін та моніторингу публічної думки про компанії.

Галузь NLP є ключовою для вирішення багатьох завдань в інформаційному суспільстві. Від аналізу тексту до розуміння емоцій та розвитку інтелектуальних систем – це поле досліджень та застосувань є важливим напрямком в розвитку сучасних технологій.

Задача розпізнавання тональності тексту за допомогою датасету STT є актуальною та перспективною, особливо в контексті сучасних вимог до обробки та аналізу великих обсягів інформації в реальному часі. Застосування

сучасних методів глибокого навчання може допомогти вдосконалити результати розпізнавання та розширити сфери його застосування.

Реалізація створеної нейронної мережі

Обрана нейронна мережа є мережею GRU з одним вхідним шаром, одним або кількома прихованими шарами та вихідним шаром (5 нейронів, оскільки є 5 класів тональності тексту).

Переваги GRU мереж над RNN та LSTM для задачі розпізнавання тональності тексту:

1. менше параметрів: GRU має менше параметрів порівняно з LSTM. Це робить модель менш обчислювально витратною та зменшує ризик перенавчання, що є дуже важливим фактором, враховуючи дуже обмежені ресурси для навчання мережі;

2. швидше навчання: менше параметрів сприяє швидшому навчанню моделі. Важливо для завдань, де швидкість навчання та прогнозування є ключовими;

3. достатній для багатьох завдань: у багатьох завданнях GRU може забезпечити подібну або навіть кращу ефективність, ніж LSTM, при меншій складності.

4. ефективність при коротших послідовностях: GRU часто працює ефективніше за LSTM на коротших послідовностях, що може бути важливим для завдань аналізу тональності;

5. низький ризик виникнення проблеми затухання градієнту: GRU має механізми важеля та забудову, що дозволяє йому добре впоратися з проблемою затухання градієнту.

Дані із датасету STT розбиваються на відповідні класи тональності. У випадку даного завдання використовується п'ять класів: "very negative," "negative," "neutral," "positive," та "very positive."

Використовується Tokenizer з бібліотеки Keras. Він працює на основі принципу присвоєння унікального числового індексу кожному слову у тексті.

Після токенизації, отримуємо послідовності числових індексів слів, які є представленням тексту.

Вхідні дані подаються у вигляді обрізаних та вирівняних за максимальною довжиною послідовностей. Це забезпечує однаковий розмір вхідних даних для всіх прикладів.

Embedding Layer мережі забезпечує векторне представлення слів, яке враховує семантичні зв'язки між ними.

SpatialDropout1D застосовує псевдо-випадковий вихід для векторів векторного представлення, щоб уникнути перенавчання.

GRU Layer: рекурентний шар для врахування контексту послідовностей.

Dense Layers: повністю з'єднані шари з активацією "relu" для вирішення завдання класифікації. Останній Dense Layer має 5 виходів (кількість класів) і активацію "softmax" для отримання ймовірностей кожного класу.

Для розрахунку втрат використовується функція "categorical_crossentropy," оскільки маємо справу з задачею класифікації на кілька класів.

У Dense Layers використовується функція активації "relu" для забезпечення нелінійності та уникнення лінійності в мережі. В останньому Dense Layer використовується "softmax" для отримання ймовірностей.

Архітектура мережі розроблена для ефективного розпізнавання тональності тексту, використовуючи рекурентний шар GRU та нейронні мережі глибокого навчання. Токенайзер та обробка даних гарантують правильну підготовку тексту для навчання та оцінки.

Для реалізації нейронної мережі використовується бібліотека TensorFlow з використанням Keras API. Код розбитий на етапи, включаючи завантаження даних, побудову моделі, навчання та оцінку результатів. Застосовується звичайна процедура тренування за допомогою стохастичного градієнтного спуску та зворотнього поширення помилки.

Для практичної реалізації створеної нейронної мережі використовується потужна бібліотека TensorFlow, яка забезпечує ефективні інструменти для створення, навчання та експериментування з нейронними мережами. Реалізація проводиться через високорівневий інтерфейс Keras API, який дозволяє легко створювати та конфігурувати нейронні мережі.

Після тренування мережі було проведено декілька тестів із випадковими фразами із тестового набору даних. Результати можна побачити на рис. 1, 2.

```
1/1 [=====] - 0s 15ms/step
INFO:root:raw prediction for <into melodrama and silliness>
[[0.02238853 0.18061306 0.45837063 0.27964297 0.05898478]]
INFO:root:Text <into melodrama and silliness>:
Predicted tonality raw: [[0.02238853 0.18061306 0.45837063 0.27964297 0.05898478]];
actual tonality raw: [0. 0. 1. 0. 0.]
Predicted tonality neutral; actual tonality: neutral
```

Рисунок 1 – Позитивний результат тестування мережі

```
1/1 [=====] - 0s 17ms/step
INFO:root:raw prediction for <The script becomes lifeless and falls apart like a cheap lawn
chair .>
[[0.01509116 0.09251859 0.3684788 0.3783191 0.14559232]]
INFO:root:Text <The script becomes lifeless and falls apart like a cheap lawn chair .>:
Predicted tonality raw: [[0.01509116 0.09251859 0.3684788 0.3783191 0.14559232]];
actual tonality raw: [0. 1. 0. 0. 0.]
Predicted tonality positive; actual tonality: negative
```

Рисунок 2 – Негативний результат тестування мережі

Висновок

Застосування звичайної GRU для розпізнавання тональності тексту на базі набору даних STT виявилось ефективним та практично здійсненим підходом. Виділення ключових методичних аспектів, аналіз предметної галузі та створення детального опису нейронної мережі дозволили створити модель, яка здатна до точного та швидкого розпізнавання цифр.

Результати експериментів та оцінки моделі на тестовому наборі відображені в розділі 4 цієї роботи, підтверджують достатньо високу ефективність застосованого підходу. Точність класифікації та інші метрики підтверджують здатність навченої мережі правильно розпізнавати рукописні цифри з задовільною точністю.

Здобуті знання та результати можуть служити основою для подальших досліджень у галузі розпізнавання образів та розвитку більш складних моделей. Дана робота відкриває нові можливості для впровадження подібних систем у сучасних технологічних рішеннях в сферах розваг, освіти, медицини та інших, де важливим є оцінити стан та настрої клієнта за текстом.

Список джерел

1. Головна сторінка документації Python-telegram-bot.: [Електронний ресурс]. – Режим доступу: <https://docs.python-telegram-bot.org/en/v20.7/>
2. Головна сторінка документації Tensorflow.: [Електронний ресурс]. – Режим доступу: https://www.tensorflow.org/api_docs/python/tf
3. Загальний огляд Keras API.: [Електронний ресурс]. – Режим доступу: <https://www.tensorflow.org/guide/keras>
4. Каллан, Р. Нейронні мережі: Короткий довідник / Р. Каллан. - М.: Вільямс І.Д., 2017. - 288 с
5. Опис набору даних Stanford sentiment treebank.: [Електронний ресурс]. – Режим доступу: <https://nlp.stanford.edu/sentiment/>

ДОСЛІДЖЕННЯ ТА АНАЛІЗ ВИКОРИСТАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ РУКОПИСНИХ ЦИФР

Халимон В.К.

Харківський національний автомобільно-дорожній університет

У сучасному світі нейронні мережі стали неот'ємною частиною багатьох сфер, включаючи обробку зображень та розпізнавання образів. Однією з ключових задач є розпізнавання рукописних цифр, що може бути вирішено за допомогою нейронних мереж. У даній курсовій роботі буде розглянуто застосування звичайної MLP (Multilayer Perceptron) для розпізнавання цифр на основі відомого набору даних MNIST.

Розвиток нейронних мереж та їх використання в різних областях життя в останні роки значно зростає. Однією з ключових областей, де виявляється великий потенціал нейронних мереж, є розпізнавання образів. У цьому контексті особливий інтерес представляє завдання розпізнавання рукописних цифр.

З впровадженням технологій машинного навчання стає можливим вирішення цього завдання за допомогою нейронних мереж, зокрема, Multi-Layer Perceptron (MLP) або звернутої мережі. Одним із стандартних наборів даних для такого роду завдань є MNIST (Modified National Institute of Standards and Technology), що містить рукописні зображення цифр від 0 до 9.

Розпізнавання рукописних цифр має широкі застосування в сучасному світі, починаючи від систем автоматичного сортування пошти та закінчуючи розпізнаванням текстів на медичних зображеннях. Використання нейронних мереж у цьому контексті дозволяє створювати ефективні та точні системи, які можуть забезпечити високий рівень автоматизації у розпізнаванні рукописних символів.

Характеристика об'єкта дослідження

Для реалізації розпізнавання рукописних цифр буде використано набір даних MNIST, який містить 60 тисяч тренувальних та 10 тисяч тестових зображень рукописних цифр. Описаний підхід використовує нейронну мережу зі стандартними шарами (вхідний, прихований, вихідний) та функцією активації для кожного шару.

MNIST включає чорно-білі зображення розміром 28x28 пікселів, представлені у вигляді матриці пікселів.



Рисунок 1 – Приклад набору MNIST

Набір даних MNIST (Modified National Institute of Standards and Technology) став фундаментальним об'єктом для багатьох досліджень у галузі машинного навчання та комп'ютерного бачення. Створений для ефективно оцінки та порівняння алгоритмів розпізнавання образів, він включає 70 тисяч чорно-білих зображень рукописних цифр, які розміром 28x28 пікселів. Цей набір даних відзначається своєю простотою та однообразністю, що робить його ідеальним об'єктом дослідження для вивчення та розробки моделей машинного навчання.

Характеристики зображень

Кожне зображення у наборі MNIST представлене у форматі матриці пікселів, де кожен піксель визначає інтенсивність чорного або білого кольору. Розмір кожного зображення складає 28 на 28 пікселів, утворюючи область 784 пікселів. Така структура дозволяє представити кожен цифру у вигляді вектора, який може бути подано нейронній мережі в якості вхідного сигналу для навчання та тестування.

Задача розпізнавання рукописних цифр

Головною задачею у цьому дослідженні є розпізнавання рукописних цифр за допомогою нейронної мережі. Кожній цифрі від 0 до 9 призначено відповідна мітка, і модель навчається класифікувати цифри на основі зразків, які представлені у наборі даних. Це завдання становить виклик для нейронних мереж, оскільки вони повинні навчитися визначати унікальні особливості кожної цифри та генерувати відповіді, що максимально наближені до дійсних значень.

Опис обраної нейронної мережі

Обрана нейронна мережа є мережею MLP з одним вхідним шаром (784 нейрони, що відповідають розміру зображення), одним або кількома прихованими шарами та вихідним шаром (10 нейронів, оскільки є 10 класів цифр). Використовується функція активації ReLU для прихованих шарів та softmax для вихідного шару.

Архітектура MLP (Multilayer Perceptron)

Обрана нейронна мережа базується на звичайній архітектурі Multilayer Perceptron (MLP), яка визначається своєю здатністю до навчання та адаптації до складних взаємозв'язків у вхідних даних. MLP складається з трьох типів шарів: вхідний, прихований і вихідний.

Вхідний шар

Вхідний шар має 784 нейрони, що відповідають розміру зображення, яке подається на вхід нейронній мережі. Кожен нейрон в цьому шарі представляє піксель зображення розміром 28x28 пікселів, яке плоско розглядається.

Приховані шари

Мережа може містити один або декілька прихованих шарів, де кожен нейрон в прихованому шарі взаємодіє з кожним нейроном попереднього та наступного шарів. Використовуються функції активації ReLU (Rectified Linear Unit) для прихованих шарів, які надають можливість моделі виявляти та вивчати нелінійні залежності в даних.

Вихідний шар

Вихідний шар має 10 нейронів, оскільки задача розпізнавання рукописних цифр полягає в класифікації на 10 категорій (цифри від 0 до 9). Для отримання ймовірностей кожного класу використовується функція активації softmax, яка перетворює виходи нейронів в ймовірності.

Застосування функцій активації

Функції активації ReLU для прихованих шарів дозволяють уникнути проблеми ванішінга градієнта, забезпечуючи ефективне навчання мережі. Функція активації softmax вихідного шару перетворює вихідні значення в ймовірності та робить модель придатною для многокласової класифікації.

Загальна концепція

Отже, обрана архітектура нейронної мережі створює просторову модель для вивчення складних взаємодій у вхідних даних та ефективної класифікації рукописних цифр. Підбір параметрів та функцій активації є ключовим для досягнення високої точності розпізнавання в межах даної задачі.

Реалізація створеної нейронної мережі

Для реалізації нейронної мережі використовується бібліотека TensorFlow з використанням Keras API. Код розбитий на етапи, включаючи завантаження даних, побудову моделі, навчання та оцінку результатів. Застосовується звичайна процедура тренування за допомогою стохастичного градієнтного спуску та зворотнього поширення помилки.

Для практичної реалізації створеної нейронної мережі використовується потужна бібліотека TensorFlow, яка забезпечує ефективні інструменти для створення, навчання та експериментування з нейронними мережами. Реалізація проводиться через високорівневий інтерфейс Keras API, який дозволяє створювати та конфігурувати нейронні мережі з легкістю.

Основні етапи включають:

1. Завантаження даних: В цьому етапі використовуються інструменти TensorFlow для завантаження набору даних MNIST, які використовуються для тренування та тестування моделі.

2. Побудова моделі: Використовуючи Keras API, створюється архітектура нейронної мережі з вхідним, прихованим та вихідним шарами. Параметри моделі, такі як кількість шарів, кількість нейронів та функції активації, конфігуруються в цьому етапі.

3. Навчання моделі: Використовуючи стохастичний градієнтний спуск та зворотне поширення помилки, модель навчається на тренувальному наборі даних. Ваги нейронів оновлюються так, щоб мінімізувати функцію втрат.

4. Оцінка результатів: Після завершення тренування проводиться оцінка результатів на тестовому наборі даних. Використовуються різні метрики, такі як точність класифікації та матриця плутанини, для об'єктивного визначення ефективності нейронної мережі.

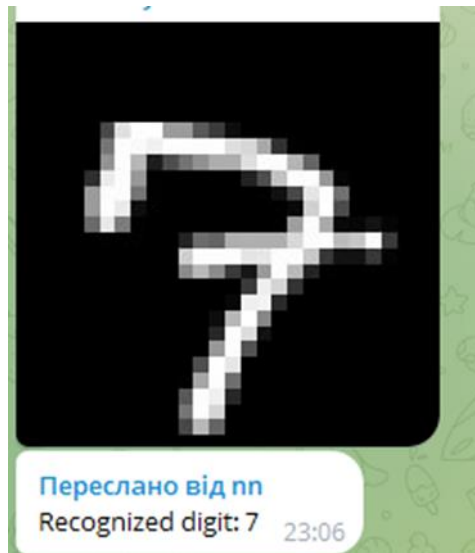


Рисунок 2 – Виконання запиту

Висновок

У даній курсовій роботі було розглянуто застосування нейронних мереж для розпізнавання рукописних цифр, проведено глибокий аналіз та практичну реалізацію нейронних мереж для розпізнавання рукописних цифр, зосереджуючись на використанні набору даних MNIST та звичайної архітектури MLP. Отримані результати та виявлені в ході експериментів особливості мають значущий вплив на розвиток та впровадження систем розпізнавання образів у різноманітних областях застосування.

Список джерел

1. Шуклін Д. Є. Моделі семантичних нейронних мереж та їх застосування в системах штучного інтелекту: 05.13.23.. Дис.канд. техн. наук. Харків, 2003. 196 с.
2. Schmidhuber, J. Deep Learning in Neural Networks: An Overview // Neural Networks. – 2015. – Vol. 61. – P. 85–117.

ВИКОРИСТАННЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ СЕНТИМЕНТ-АНАЛІЗУ

Харченко С. Д.

Харківський національний автомобільно-дорожній університет

В аналізі текстів, зокрема в визначенні тональності, постійно розвиваються та вдосконалюються методи та підходи. Запитання про те, як автоматизовано розпізнавати емоційний відтінок висловлених думок та висловлювань, є важливим завданням у сучасній обробці природної мови. В даному розділі ми ретельно розглянемо сучасні методи визначення тональності, починаючи від традиційних правилкових підходів та евристик, переходячи до методів машинного навчання та фокусуємось на застосуванні нейронних мереж.

Правила та евристичні методи визначення тональності використовують заздалегідь визначені правила та словники для ідентифікації позитивних чи негативних емоцій у текстах. З одного боку, це ефективні методи для простих сценаріїв, але вони обмежені у врахуванні контексту та адаптації до різних стилів мови.

Методи машинного навчання дозволяють автоматично визначати емоційний тон текстів, використовуючи моделі, навчені на великих наборах позначених даних. Ці підходи виявилися ефективними у вирішенні завдань аналізу тональності, але вони також мають свої обмеження, зокрема, у врахуванні складних залежностей в текстах та управлінні суб'єктивністю.

Останнім часом нейронні мережі завоювали популярність завдяки своїй здатності автоматично вивчати складні залежності та пристосовуватися до різних умов. У цьому контексті вони стають потужним інструментом для аналізу тональності, дозволяючи враховувати контекст, адаптуватися до різних стилів мови та працювати з великими обсягами даних.

Визначення методу для перевірки тональності тексту

МЛ-тональність - механізм визначення тональності тексту стосовно об'єкта моніторингу за допомогою AI-моделі, натренованої на корпусі текстів. Ці тексти спочатку були розмічені вручну за тональністю по відношенню до об'єктів, що зустрічалися в текстах. Для цього використовувались правила оцінки тональності, які використовуються під час роботи медіа-аналітиків.

Для МЛ-тональності використовується так званий об'єктно-орієнтований алгоритм визначення тональності (Entity related sentiment analysis), тобто емоційне забарвлення повідомлень визначається не просто для публікації як такої, а стосовно об'єкту моніторингу у темі. Об'єкт моніторингу – це ті ключові слова, які у вашій темі/темах визначені як "виділені ключові слова".

Під час створення теми менеджером системи або користувачем необхідно в окреме поле виділених ключових слів ввести ті слова та словосполучення, відносно яких буде визначатись тональність.

Так як для роботи були обрані згорткові нейронні мережі, нижче наведемо алгоритм їх дії. Алгоритм роботи з загортковими нейронними мережами:

1. Підготовка даних:

Перетворення тексту в числовий вектор за допомогою токенизації та векторизації слів (наприклад, за допомогою векторизаторів, як TF-IDF чи Word2Vec). Поділ даних на тренувальний та тестовий набори.

2. Створення моделі згорткової нейронної мережі:

Створення шарів згортки та пулінгу для виділення важливих ознак у тексті. Використання рекурентних шарів або шарів LSTM (Long Short-Term Memory) для збереження контексту між словами. Використання повнозв'язаних шарів для класифікації тексту на позитивний, негативний чи нейтральний.

3. Навчання та оптимізація:

Використання тренувального набору для навчання моделі. Вибір функції втрат та оптимізатора для мінімізації втрат під час навчання. Налаштування гіперпараметрів, таких як розмір фільтрів та кількість шарів.

4. Тестування та оцінка:

Використання тестового набору для оцінки продуктивності моделі. Визначення метрик ефективності, таких як точність, відновлення, та F-мера.

5. Удосконалення моделі:

Вирішення проблем перенавчання шляхом використання методів регуляризації та викидів. Оптимізація гіперпараметрів для покращення результатів.

Метод Adam

Вибір алгоритму для оптимізації системи є дуже важливим питанням, адже це може певним чином покращити результати навчання, та час навчання мережі.

Метод Адама є досить новим, та отримав досить велику популярність при навчанні глибоких мереж у сферах комп'ютерного зору та розпізнавання мови та тексту. Також варто зазначити що цей метод є модифікацією методу стохастичного градієнтного спуску, тож часто використовується замість нього для ітеративного корегування ваг мережі на основі вхідних даних.

Розглянемо чому саме обирають цей алгоритм, для цього наведемо його переваги котрі надали автори методу:

- Метод добре працює із задачами, котрі вимагають великих об'ємів даних;
- Є досить простим та не привередливим до сервера на котрому відбувається навчання, хоча б тому що не потребує багато пам'яті;
- Гіперпараметри методу є досить простими для розуміння та часто не потребують важкого та довгого налаштування;
- Відносно інших модифікацій є досить простим для реалізації та не потребує довгого написання коду;
- Має велику обчислювальну ефективність;

Порівнюючи метод Adam з його початковою версією – методом стохастичного градієнтного спуску, то перший підтримує постійну швидкість

навчання для всіх регуляцій ваг, в другому випадку було поєднано найкращі сторони двох інших модифікацій першого і метод адаптує швидкість навчання для кожного з параметрів використовуючи першого та другого моментів градієнту. Виходить так, що Adam на відміну від методу RMSProP використовує не тільки середнє першого моменту, але й середнє другого моменту градієнта – нецентрована дисперсія.

Для демонстрації можливостей цього методу було проведено навчання багатошарового перцептрона і порівняно результати(training cost) з іншими методами. Далі розглянемо параметри котрі має даний метод, вони мають певні стандартні значення у популярних бібліотеках наприклад мови Python, таких як Tensorflow, Keras(надбудова Tensorflow), Blocks.

- Параметр α - це значення задає швидкість навчання, або розмір кроку з котрим оновлюються ваги.

- Параметр b_1 – це показник експоненційного затухання першого моменту

- Параметр b_2 – цей параметр має такий самий сенс як і попередній але для другого моменту, це значення може бути близьким до 1 або навпаки дуже малим значенням. Всі ці параметри мають певні стандартні та рекомендовані значення у вище наведених бібліотеках для спрощення роботи з ними.

Опис обраної нейронної мережі

Нейромережа, написана на Python, буде відрізнитися швидкістю, безпекою та стабільністю. Сама мова програмування передбачає такі переваги: функціональність, поділ підсумкової кодифікації на блоки, що дозволяє значно підвищити її читання, підтримка довгої арифметики, кросплатформність, безліч бібліотек, які зможуть отримати в будь-який час, зрозумілий синтаксис. Це ідеальний варіант для веб-розробки, додатків для месенджерів та дрібних проектів.

Sequential (tensorflow.keras.models.Sequential) – Ця бібліотека надає можливість створення лінійних моделей шляхом послідовного додавання шарів. Використовується для зручного визначення та навчання нейронних мереж.

Dense (tensorflow.keras.layers.Dense) – Шар, що представляє повнозв'язаний шар у нейронній мережі. Кожен нейрон у цьому шарі пов'язаний з кожним нейроном попереднього та наступного шарів.

Embedding (tensorflow.keras.layers.Embedding) – Шар вбудування для конвертації текстових даних в числовий вектор, де кожне слово представлено унікальним числовим значенням.

Conv1D (tensorflow.keras.layers.Conv1D) – Шар одновимірної згорткової нейронної мережі, який використовується для виявлення локальних залежностей у текстових даних.

GlobalMaxPooling1D (tensorflow.keras.layers.GlobalMaxPooling1D) – Шар для вибору максимального значення з кожного згорткового фільтра, що допомагає визначити найважливіші ознаки.

`Dropout (tensorflow.keras.layers.Dropout)` – Шар випадкового відключення, який допомагає уникнути перенавчання за допомогою випадкового вимкнення деяких нейронів під час навчання.

`pad_sequences (tensorflow.keras.preprocessing.sequence.pad_sequences)` – Функція для доповнення або обрізання послідовностей (текстових даних) до заданої довжини. Використовується для створення вхідних даних фіксованого розміру для моделі.

`Tokenizer (tensorflow.keras.preprocessing.text.Tokenizer)` – Клас для токенизації текстових даних, перетворення слів у числові індекси та створення словників для подальшого використання у вбудуванні.

Реалізація створеної нейронної мережі

Для реалізації нейронної мережі використовується бібліотека TensorFlow з використанням Keras API. Код розбитий на етапи, включаючи завантаження даних, побудову моделі, навчання та оцінку результатів. Застосовується звичайна процедура тренування за допомогою стохастичного градієнтного спуску та зворотнього поширення помилки.

Для практичної реалізації створеної нейронної мережі використовується потужна бібліотека TensorFlow, яка забезпечує ефективні інструменти для створення, навчання та експериментування з нейронними мережами. Реалізація проводиться через високорівневий інтерфейс Keras API, який дозволяє створювати та конфігурувати нейронні мережі з легкістю.

Основні етапи включають:

1. Завантаження даних: В цьому етапі використовуються інструменти TensorFlow для завантаження набору даних MNIST, які використовуються для тренування та тестування моделі.

2. Побудова моделі: Використовуючи Keras API, створюється архітектура нейронної мережі з вхідним, прихованим та вихідним шарами. Параметри моделі, такі як кількість шарів, кількість нейронів та функції активації, конфігуруються в цьому етапі.

3. Навчання моделі: Використовуючи стохастичний градієнтний спуск та зворотнє поширення помилки, модель навчається на тренувальному наборі даних. Ваги нейронів оновлюються так, щоб мінімізувати функцію втрат.

4. Оцінка результатів: Після завершення тренування проводиться оцінка результатів на тестовому наборі даних. Використовуються різні метрики, такі як точність класифікації та матриця плутанини, для об'єктивного визначення ефективності нейронної мережі.

Висновок

У даному проекті я використовувала бібліотеки Python, зокрема Pandas для обробки та аналізу табличних даних. Моя робота включала в себе етапи завантаження даних, їх підготовку та взаємодію з ними. Я вивчала та аналізу-

вала набір даних за допомогою Pandas, використовуючи методи для вибірки, фільтрації, та обчислення агрегованих статистик. Важливим етапом було використання бібліотеки Matplotlib для візуалізації результатів та отримання більш глибокого розуміння даних.

Під час проекту я також використовувала нейромережі, зокрема Tensor Flow та Keras, для навчання моделі на текстових даних. Я завантажувала та підготовлювала дані, використовуючи Tokenizer для векторизації тексту. Моя робота включала створення та навчання моделі, аналіз її результатів та тестування на нових даних.

Усе це спрямовано на досягнення кращого розуміння даних, виявлення закономірностей та навчання моделей для ефективного вирішення конкретних завдань у сферах обробки даних та машинного навчання.

У роботі з аналізом тональності тексту нейронні мережі виявляються потужним та ефективним інструментом. Застосування цих алгоритмів до визначення емоційного відтінку висловлень не лише дозволяє автоматизувати процес, але і забезпечує значно кращі результати в порівнянні з традиційними методами. Згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), використовуючи свою здатність враховувати контекст та вивчати складні залежності в тексті, стали ключовими гравцями у вирішенні задачі визначення тональності. Використання шарів згортки та пулінгу дозволяє виділяти ключові ознаки, тоді як рекурентні шари враховують послідовність слів. Процес обробки текстових даних перед використанням в нейронних мережах також важливий. Використання Tokenizer для перетворення текстових відгуків у числові послідовності дозволяє нам подати інформацію так, що її можна використовувати в якості вхідних даних для моделей. Застосування нейронних мереж в аналізі тональності дозволяє не лише отримувати точні результати, але і адаптуватися до різних стилів мови, враховувати контекст та працювати з великими обсягами даних. Це робить їх особливо корисними в сучасному середовищі, де обробка та розуміння великої кількості текстової інформації є ключовими завданнями.

Необхідно відзначити, що використання нейронних мереж вимагає належного підготовчого етапу, великої кількості даних для навчання та оптимізації гіперпараметрів. Проте, з усіма викликами, вони стають потужним засобом для вирішення завдань визначення тональності тексту в різних сферах, таких як відгуки користувачів, аналіз соціальних мереж, та багато інших.

Список джерел

1. Аналіз тональності текстів за допомогою згорткових нейронних мереж.: [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/companies/vk/articles/417767/>
2. An easy tutorial about Sentiment Analysis with Deep Learning and Keras.: [Електронний ресурс]. – Режим доступу: <https://towardsdatascience.com/an-easy-tutorial-about-sentiment-analysis-with-deep-learning-and-keras-2bf52b9c91>
3. How To Train a Neural Network for Sentiment Analysis.: [Електронний ресурс]. – Режим доступу: <https://www.digitalocean.com/community/tutorials/how-to-train-a-neural-network-for-sentiment-analysis>

РОЗРОБКА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ СЕГМЕНТАЦІЇ ОБ'ЄКТІВ ДОРОЖНЬОГО РУХУ

Черняк О.К.

Харківський національний автомобільно-дорожній університет

Практична реалізація створеної нейронної мережі для сегментації об'єктів дорожнього руху може включати в себе такі етапи, які використовують бібліотеку TensorFlow з використанням Keras API:

1. Імпорт бібліотек:
 - Включення необхідних бібліотек, таких як TensorFlow, Telebot та Keras, у проект.
2. Підготовка даних:
 - Завантаження та підготовка набору даних для навчання та валідації.
 - Аугментація зображень для підвищення робастності мережі.
3. Створення моделі:
 - Використання Keras для визначення архітектури MLP за допомогою Sequential або Functional API.
 - Додавання вхідного та прихованих шарів з використанням Dense шарів.
 - Встановлення функцій активації для прихованих шарів та softmax для вихідного шару.
 - Компіляція моделі з обраною функцією втрат та оптимізатором.
4. Навчання моделі:
 - Використання методу `fit` для тренування моделі на навчальних даних.
 - Передача валідаційних даних для моніторингу ефективності та уникнення перенавчання.
 - Встановлення кількості епох та інших гіперпараметрів для оптимального навчання.
5. Інтеграція у телеграм-бота:
 - Розробка телеграм-бота з використанням Telegram Bot API.
 - Налаштування інтерфейсу бота для взаємодії з моделлю: відправлення та отримання зображень, обробка запитань користувача.
6. Тестування та оцінка результатів:
 - Застосування моделі до тестового набору даних для оцінки її точності та загальної ефективності.
 - Моніторинг метрик, таких як точність та втрати.
7. Оптимізація та налаштування:
 - В разі необхідності внесення змін до архітектури, гіперпараметрів або процесу навчання для покращення результатів.
8. Документація та звітність:
 - Створення документації, яка описує кожен етап розробки та параметри моделі.
 - Підготовка звіту, включаючи аналіз результатів та висновки.

Ці кроки представляють загальний підхід до практичної реалізації нейронної мережі для сегментації об'єктів дорожнього руху та її інтеграції у телеграм-бота з використанням TensorFlow та Keras.

Висновок

Отримані результати свідчать про успішність застосування нейронних мереж у сегментації об'єктів дорожнього руху. Проте, для поліпшення точності та робастності моделі можливі подальші дослідження та оптимізації. Також, розширення функціональності телеграм-бота та взаємодія з користувачем може бути розглянуто у майбутньому. В цілому, дана робота вносить свій внесок у розвиток технологій комп'ютерного зору та застосування нейронних мереж у сфері розпізнавання об'єктів дорожнього руху.

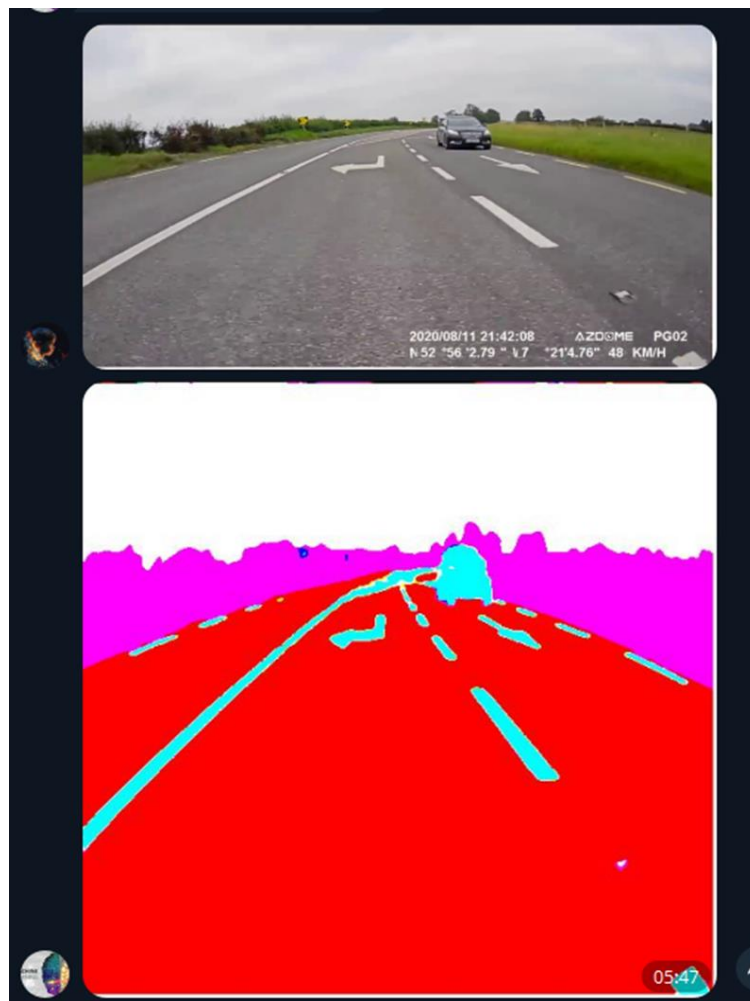


Рисунок 1 – Тестування проєкту

Список джерел

1. Шуклін Д. Є. Моделі семантичних нейронних мереж та їх застосування в системах штучного інтелекту: 05.13.23.. Дис.канд. техн. наук. Харків, 2003. 196 с.
2. Schmidhuber, J. Deep Learning in Neural Networks: An Overview // Neural Networks. – 2015. – Vol. 61. – P. 85–117.

ВИКОРИСТАННЯ ШІ У СУЧАСНОМУ ДИЗАЙНІ

Шніть Єгор, Пронін С.В.

Харківський національний автомобільно дорожній університет

Сьогодні я хотів би розповісти вам про використання нейронних мереж у графічному редакторі Photoshop.

Сама по собі нейронна мережа – це математична модель, а також її програмна або апаратна реалізація, побудована за принципом організації та функціонування біологічних нейронних мереж - мереж нервових клітин живого організму. з'являється друга індустрія, якою користується більшість шахраїв, а саме підміна облич в будь-якому відео треку або навіть в прямих ефірах.

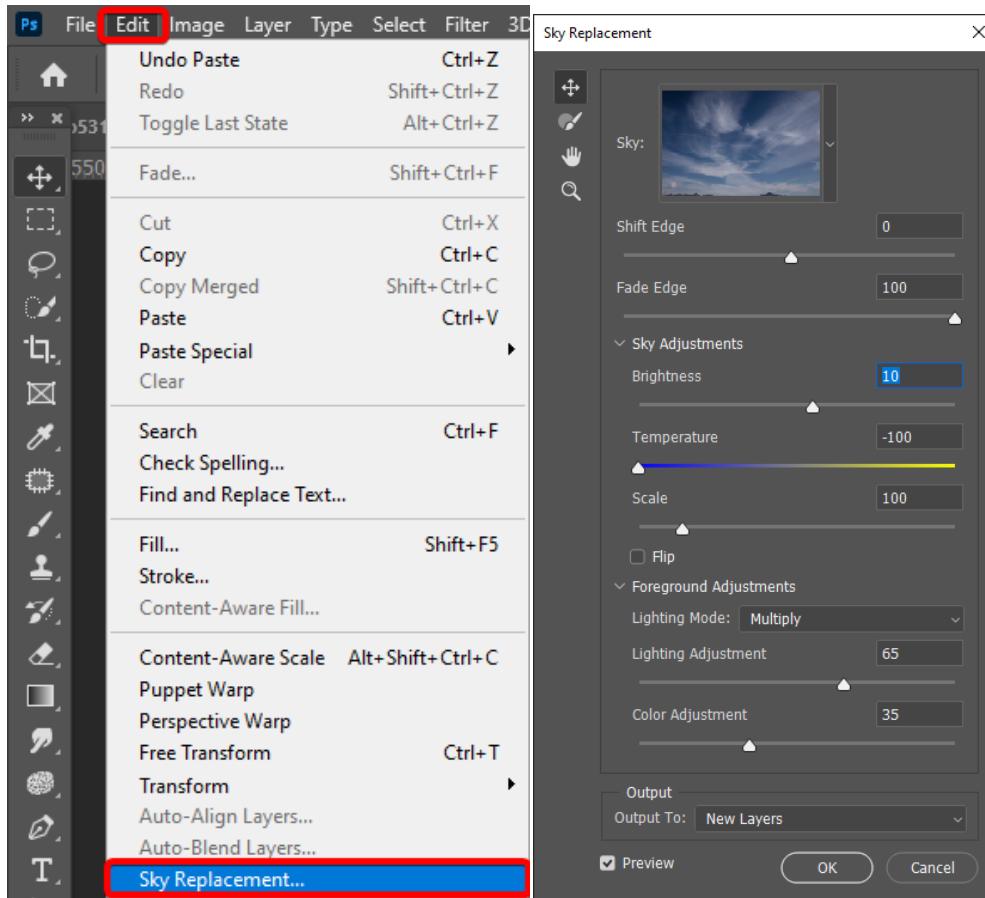
Але сьогодні я розповім вам про безпечні нейронні мережі, які вже не дуже активно починають впроваджуватися в один з найпопулярніших графічних редакторів Adobe Photoshop на 2024 рік. В даному випадку нейромережі налаштовуються для полегшення роботи в конкретних випадках. Кому це потрібно? досвід і йому потрібно щось зробити терміново і швидко, навіть не переглядаючи ніяких навчальних відео. Якщо ви людина, яка давно сидить в сфері обробки фотографій, для вас це скорочення часу на той чи інший процес.

(Відразу хочу попередити, що це тестові версії нейронних мереж в фотошопі і з кожним роком їх буде все більше, тому не все буде працювати ідеально зараз) Розглянемо пару фільтрів як приклад, а саме: заміна часу доби за допомогою неба, додавання ефекту димки дальності на фотографії, розфарбування чорно-білої фотографії, видалення кольорових мертвих пікселів через погану камеру, перенесення макіяжу і фільтрація з людськими змінами на фотографії.

Ми почнемо з найпростішого фільтра, Заміна неба, і це звичайне зображення, взяте з інтернету.



Далі ми просто переходимо до редагування і зі спливаючого списку натискаємо "Замінити небо".

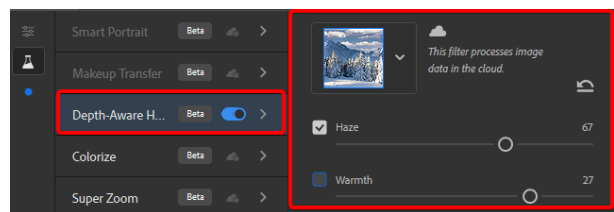


Цей нейрофільтр самостійно підбирає маску неба і, в залежності від неба, світло-тіньовий малюнок буде змінюватися, як у прикладі нижче. (Вже оброблене фото)



(Всі наступні нейрофільтри відображалися на 1 панелі, куди ви можете увійти, натиснувши на фільтр і у спливаючому списку натиснути на Нейронні фільтри) Тепер ми залишимося на тій самій фотографії і застосуємо до неї нейрофільтр "серпанок з глибиною".

Результат до цього:

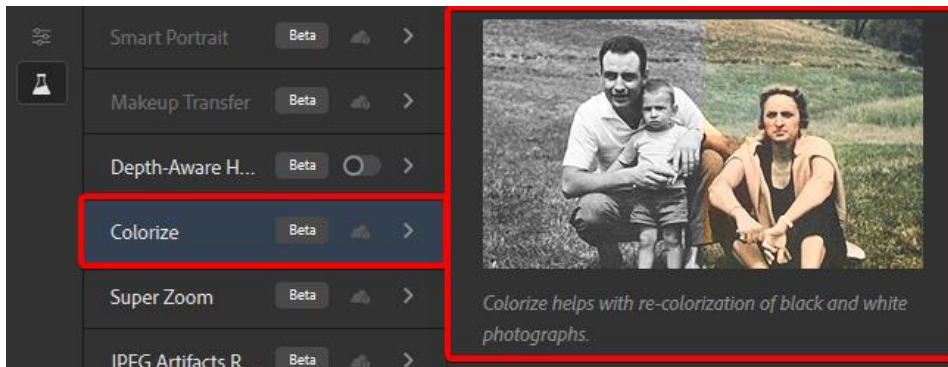


Результат після:



Як бачимо, цей фільтр дозволяє сфокусуватися на конкретному об'єкті (серпанок накладається на весь шар, тому вам доведеться працювати з маскою далі, тому він не підійде для недосвідчених людей)

На черзі нейрофільтр, який дозволяє оживити фотографії, зроблені в далекі роки на чорно-білу плівку.



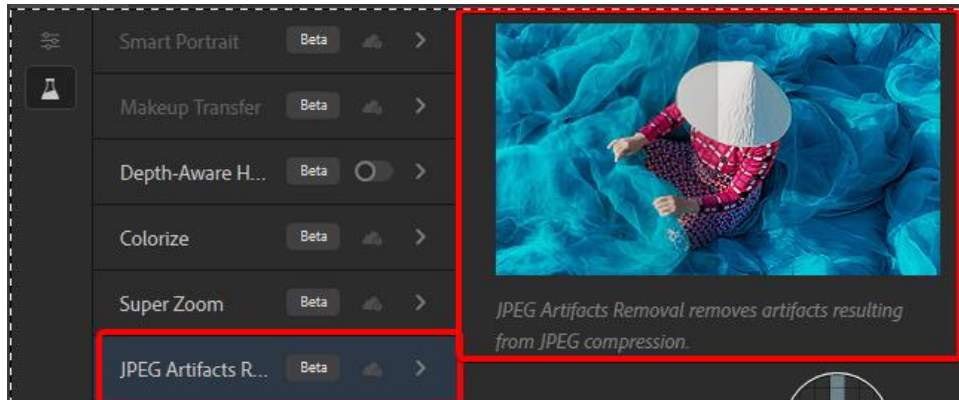
Беремо аналогічне зображення:



Потім ми застосовуємо фільтр і отримуємо наступний результат:



Переходимо до чудового фільтру для фотографів-початківців з поганою камерою під назвою "Видалення дефектів". Цей фільтр усуває дефекти, застосовуючи легке розмиття до певних ділянок зображення.



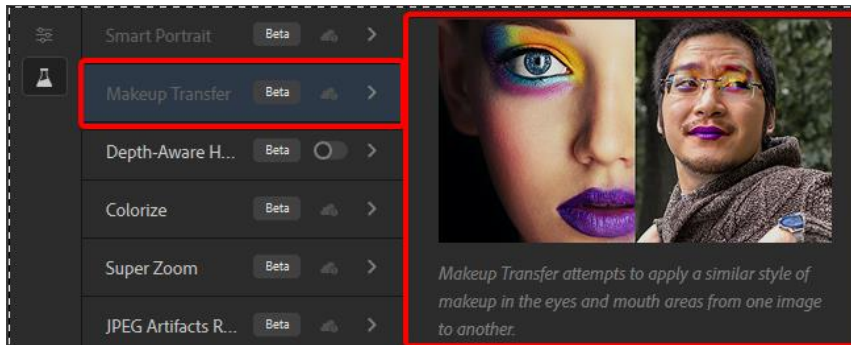
Попередній результат:



Результат після:



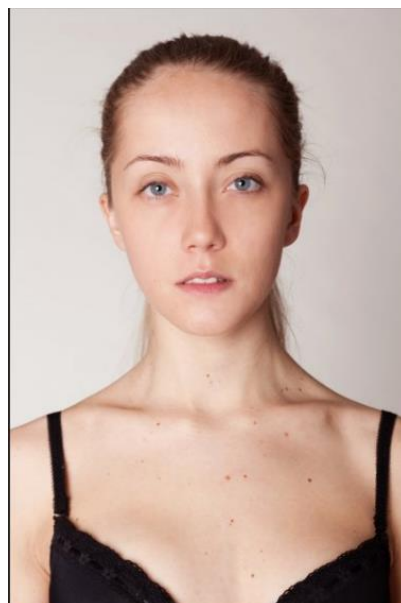
Тепер давайте розглянемо фільтр для наших прекрасних дівчат. Вашій дружині дуже сподобався макіяж на якійсь із зірок, але у неї немає такого величезного асортименту косметики для цього, ставимо її перед камерою, фотографуємо і застосовуємо цей фільтр.



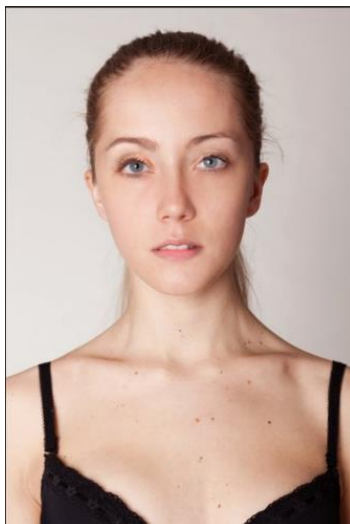
Беремо фото з макіяжем:



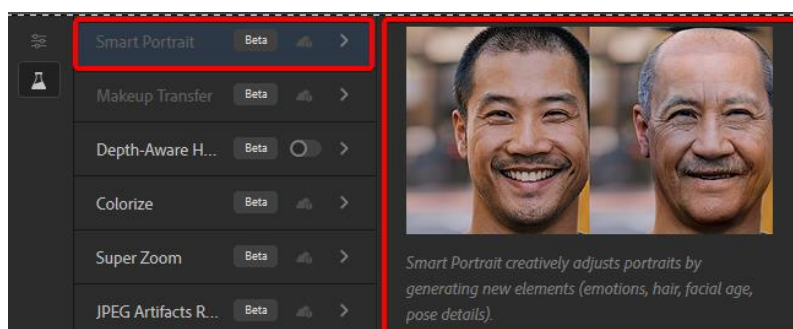
Беремо фото без макіяжу:



Отримуємо фінальний результат:



Ну і в моїй розповіді залишився тільки 1 фільтр, про який я нічого не сказала, а саме фільтр, який дозволяє змінити емоцію на фото. Розглянемо пару прикладів, жінка прийшла на фотосесію в модельне агентство вже у віці і вона не може показати посмішку і ви взагалі хотіли бачити інший результат, але якщо це єдиний шанс і у вас немає часу змінити цей фільтр для вас.



Попередній результат:



Результат після:



Боже мій, ви навіть можете повернути їй голову!!! (не працює ідеально)
Результат після:



На цьому я хотів би завершити свою розповідь і все ж таки зробити фінальний акцент на тому, як все ті ж нейрофільтри впливають на наше майбутнє. Нейрофільтри можуть самостійно розвиватися при їх частому використанні і на цих результатах вдосконалювати свою систему. І це тільки початок, адже в майбутньому розробники нейрофільтрів для фотошопу обіцяють такими ж простими натисканнями на пару кнопок запропонувати наступний список можливостей – реставрація фотографії ваших предків, видалення пилу і подряпин з фотографії, зменшення шумів на поганих камерах, перетворення навіть найжахливіших знімків в щось вартісне, видалення окулярів з фотографії, відтворення з фотографії малюнка людини і багато іншого!

У штучного інтелекта фотошопа є свої особливі переваги над іншими. А саме:

1. **Якість обробки:** нейронні фільтри дають змогу досягти високої якості обробки зображень, включно з поліпшенням деталей і кольорів, а також реалістичним ретушуванням.

2. **Зручність використання:** фільтри легко застосовуються до зображень у Photoshop і мають інтуїтивно зрозумілий інтерфейс, що робить їх доступними для широкого кола користувачів.

3. **Багатофункціональність:** нейронні фільтри надають широкий спектр можливостей для обробки зображень, що дає змогу досягти різноманітних ефектів і стилів.

4. **Сумісність:** фільтри можуть бути інтегровані з іншими інструментами та функціями Photoshop, що дає змогу створювати складні обробки та композиції зображень.

Загалом, нейронні фільтри в Photoshop є потужним інструментом для обробки зображень, який дає змогу досягти високої якості та різноманітності ефектів.

Література

1. Neural network [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Neural_network. Дата доступу: 15.04.2024

2. Photoshop [Електронний ресурс]. Режим доступу: <https://www.adobe.com>. Дата доступу: 15.04.2024

3. Редагувати зображення з генеративною заливкою [Електронний ресурс]. Режим доступу: <https://helpx.adobe.com/ua/photoshop/using/generative-fill.html>. Дата доступу: 15.04.2024

УДК 624.132.6

**РОЗРОБКА МЕТОДИЧНОГО КОМП'ЮТЕРНОГО
КОМПЛЕКСУ ПІДГОТОВКИ ОПЕРАТОРІВ ДИСПЕТЧЕРСЬКОГО
УПРАВЛІННЯ АСФАЛЬТО БЕТОННИМИ УСТАНОВКАМИ
НА ОСНОВІ ТЕХНОЛОГІЇ НМІ**

*Гриценко К.М., студент магістратури механічної кафедри
Харківського національного автомобільно-дорожнього університету*
Петренко Ю.А., д.т.н., проф.
Харківського національного автомобільно-дорожнього університету

Анотація. У тексті обговорюється важливість підготовки кваліфікованих операторів для асфальтобетонних заводів через зростаючий попит на асфальтобетонне покриття. Це підкреслює потребу в глибоких знаннях і навичках серед операторів для роботи з передовими технологіями та обладнанням, що використовується у виробництві асфальтобетону, забезпечуючи при цьому якість і ефективність.

Ключові слова: асфальтобетонні заводи, кваліфікація операторів, передові технології, виробництво асфальтобетонних сумішей

Актуальність підготовки кваліфікованих операторів асфальтобетонних заводів є дуже високою в сучасному будівництві. Попит на дорожнє покриття з бетону постійно зростає, що потребує більшої кількості фахівців, здатних експлуатувати та обслуговувати складну техніку для виробництва асфальту. Крім того, використання передових технологій та обладнання для виробництва асфальтобетону потребує більш глибоких знань та навичок з боку операторів, щоб забезпечити якість продукції та ефективність процесу.

Підготовка операторів до робіт на асфальтобетонних заводах повинна включати теоретичні основи керування матеріалами, механіку ґрунтів та будівельних конструкцій, а також практичні навички керування обладнанням та діагностику неполадок. Крім того, оператори повинні бути проінформовані про останні тенденції в галузі технологій виробництва асфальтобетону, екологічні норми та стандарти безпеки при роботі з обладнанням. Таким чином, професійна підготовка операторів сприяє підвищенню ефективності виробництва, скороченню витрат та підтримці якості кінцевого продукту, що має вирішальне значення для збереження підприємств у дорожньо-будівельній галузі.

В даний час у світі існує кілька сучасних тренажерних платформ моделювання технологічних процесів, розроблених та підтримуваних основними світовими виробниками комп'ютерних тренажерів (КТ), такими як ABB Simcon, Inc., Honeywell Hi-Spec Solutions [1].

КТ чи симулятори це імітатори, комп'ютерні чи механічні, які керують будь-яким процесом. Гнучка архітектура персональних комп'ютерів (ПК) дозволяє вирішити конфігураційні проблеми: у обчислювальній мережі сучасних КТ легко розміщуються кілька робочих місць оператора, робоча станція інструктора, інженера, технолога тощо. Сучасні КТ легко з'єднуються з різними розподільчими системами управління (DCS), програмованими логістичними контролерами (PLC), системами візуалізації даних та супервізорного управління (SCADA) та іншими засобами. Разом з тим, полегшуються і можливості застосування різних стандартів відображення інформації та управління процесом. Значний досвід використання КТ у навчальному процесі накопичено німецькою фірмою Ammann. На рис. 1 представлений навчальний центр операторів асфальтозмішувальних установок фірми Ammann. у м. Лангеталь (Швейцарія) [2]. У вітчизняній практиці навчання у ВНЗ, за винятком тренажерів для енергетики, де досягнуто якісних технічних рішень, відсутні розвинені тренажерні платформи технологічних процесів, що позначається і на якості самих моделей, і на рівні відтворення операторського інтерфейсу. Слабко представлено також методичну сторону навчання. Це характерно і під час навчання за автомобільно-дорожніми спеціальностями [1].



Рис. 1. Навчальний центр операторів асфальтозмішувальних установок

У березні 2021р. Харківський національний автомобільно-дорожній університет (ХНАДУ), приватне акціонерне товариство «Кременчуцький завод дорожніх машин» (ПрАТ «Кредмаш») та ТОВ «Науково-виробниче об'єднання «Транссистема» (ТОВ НВО «Транссистема») уклали договір про створення спільного науково-освітнього центру дорожніх машин при механічному факультеті ХНАДУ. Як показало знайомство з навчальними програмами кафедри будівельних та дорожніх машин та автоматизації та комп'ютерно-інтегрованих

технологій механічного факультету ХНАДУ, в учбовому процесі відсутні тренажери виробництва дорожньо-будівельних матеріалів [1].

ТОВ НВО «Транссистема» надало комплектуючі та розробило конструкцію навчального тренажера, що максимально імітує технологічний процес виробництва асфальтобетонних сумішей (АБС) (рис.2) [1].

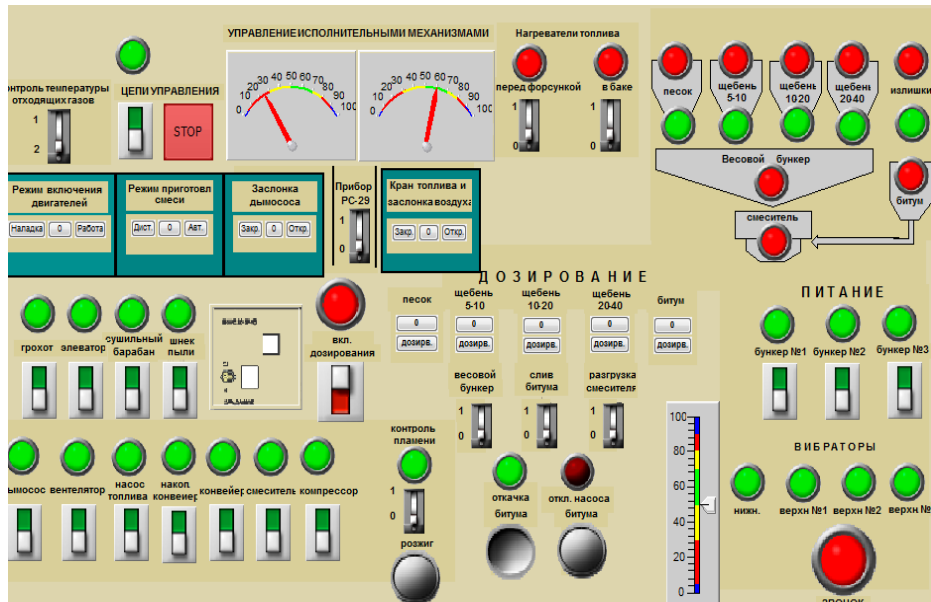


Рис. 2. Навчальний тренажер виробництва АБС

На першому етапі розробки конструкції навчального тренажера було вирішено завдання представлення випуску АБС на вітчизняних асфальто змішувальних установках, виробництва ПрАТ «Кредмаш», що використовують як мікропроцесорну систему управління, так і релейно-контактну систему управління, яка характерна для багатьох асфальто змішувальних установок, що експлуатуються в Україні [1].

В основу технічного завдання на розробку навчального тренажера було покладено навчальний посібник, розроблений ПрАТ "Кредмаш" та ТОВ НВО "Транссистема" - "Основи експлуатації асфальтозмішувальних установок" [1]. Цей навчальний посібник був створений на основі національного стандарту [4]. Значний обсяг інформації у навчальному посібнику присвячений практичним знанням з апаратного оснащення сучасних асфальтозмішувальних установок, порядку запуску та зупинки обладнання, їх технічного обслуговування та ремонту. Додатково у навчальному посібнику викладено інформацію з охорони праці та захисту навколишнього середовища під час виробництва АБС. На рис.3. представлено технологічну схему виробництва АБС з логотипами безпеки з учбового посібника [1].

На другому етапі розробки навчального тренажера зроблено узгодження пульта керування асфальтозмішувальної установки з технологічною схемою виробництва АБС із використанням сучасних тренажерних платформ моделювання технологічних процесів [1].

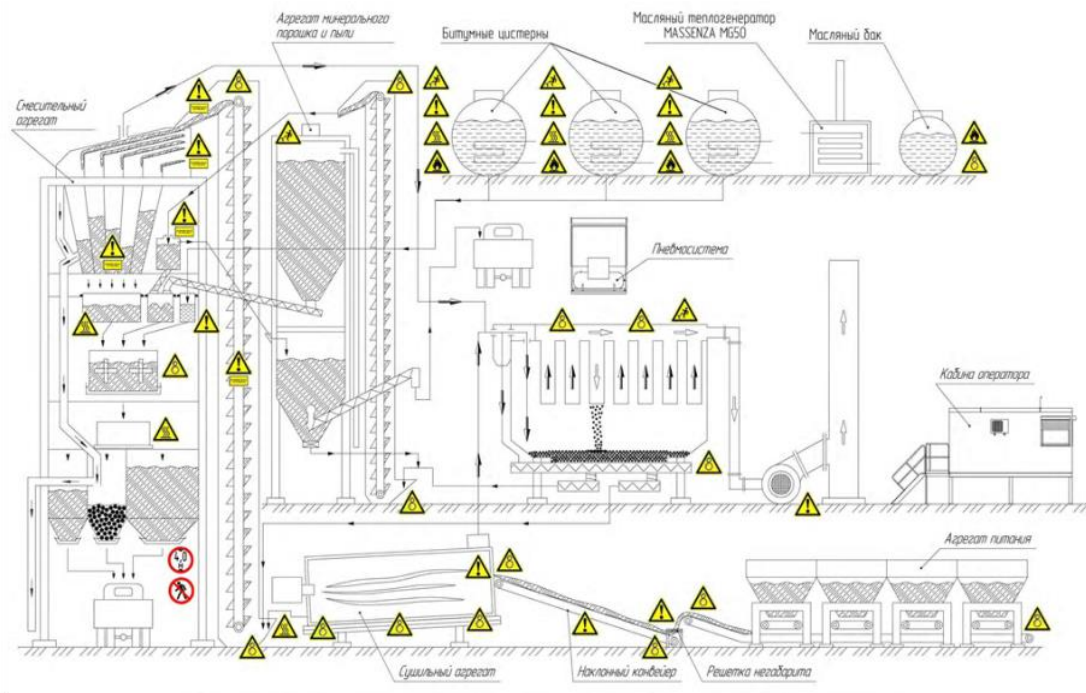


Рис. 3. Технологічна схема виробництва АБС із логотипами безпеки [1]

Додатково для візуального представлення технологічного процесу виробництва АБС ТОВ НПО «Транссистема» надало відео, що демонструє роботу агрегатів асфальтозмішувальної установки у 3D [5, 1].

Одним з основних завдань під час проведення практичних занять у процесі освоєння технології виробництва АБС є імітація дій оператора асфальтозмішувальної установки. Оператор асфальтозмішувальної установки реалізує складний та багатадійний технологічний процес, що базується на стандартах виробництва різних АБС [4, 5]. Спільно з лаборантом, який здійснює попередній підбір складу АБС, оператор повинен забезпечувати безперебійність процесу виробництва, розуміти зв'язок якості сумішей з якістю вихідних матеріалів, що обумовлює випуск якісних АБС, мінімізувати витрату палива на їх виробництво, скорочувати негативний вплив на навколишнє середовище, забезпечувати безпеку праці та подовжувати термін служби устаткування. Це значною мірою досягається під час використання у навчальному процесі навчальних тренажерів [1].

На основі аналізу технології НМІ, у майбутньому, напрямок мого дослідження, це розробка моделі вибору програмного забезпечення, для створення комп'ютерно-навчально-методичного комплексу для підготовки диспетчерів АБУ.

Література

1. Татаринський В.Б., Петренко Ю.А. Комп'ютерні тренажери виробництва асфальтобетонних сумішей. Органічні і мінеральні в'язучі та дорожні бетони на їх основі : зб. матеріалів доп. учасн. міжнародної науково-технічної конференції (8 - 9 листопада 2022 р.). – Харків: ФОП Бровін О.В., 2022. С. 156-159.

2. Офіційний сайт компанії Ammann Group. Режим доступу: info.aag@ammann-group.com.

3. Невлюдов І.Ш. Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6: Навчальний посібник / І.Ш. Невлюдов, А.О. Андрусевич, В.В. Євсєєв, С.С. Максимова, М.Г. Стародубцев, В.В. Невлюдова. – Кривий Ріг: КК НАУ, 2018. – 316 с.

4. Суміші асфальтобетонні і асфальтобетон дорожній та аеродромний. Технічні умови: ДСТУ Б В.2.7 – 119:2011 -[Чинний від 01-10-2011] – К.: Мінрегіон України, 2011. – 59 с. (Національний стандарт України).

5. Офіційний сайт компанії ЧАО «Кредмаш». Режим доступу: <http://www.kredmash.com>.

ТЕЛЕГРАМ БОТ ПОМІЧНИК ПОДОРОЖУВАЛЬНИКА

Батура Михайло Віталійович

Харківський національний автомобільно-дорожній університет, Харків

Вступ

Сучасний світ характеризується стрімким розвитком інформаційних технологій, які значно впливають на різні аспекти нашого життя, включаючи комунікацію, доступ до інформації та виконання рутинних завдань. Однією з ключових тенденцій останніх років є зростання популярності месенджерів та чат-ботів, які надають можливість автоматизувати багато процесів та зробити взаємодію з технологіями більш зручною та ефективною. Telegram, як один з провідних месенджерів, пропонує розробникам потужний інтерфейс програмування додатків (API), що дозволяє створювати інтерактивних та функціональних ботів для вирішення широкого спектру задач.

Актуальність теми

З кожним роком збільшується кількість людей, які активно подорожують, працюють у міжнародних компаніях або просто цікавляться ситуацією в інших країнах. Глобалізація, розвиток міжнародних зв'язків та збільшення мобільності населення створюють нові виклики і потреби, серед яких особливе місце займає доступ до оперативної та достовірної інформації.

Подорожі, незалежно від їх мети, вимагають ретельної підготовки. Одним з найважливіших аспектів цієї підготовки є отримання актуальної інформації про погоду в місці призначення.

Вчасно отримані дані про погодні умови дозволяють мандрівникам краще планувати свої дії, вибирати відповідний одяг, розуміти можливі погодні ризики та забезпечувати свій комфорт та безпеку. Окрім цього, актуальна інформація про валютні курси є не менш важливою. Вона допомагає орієнтуватися в місцевій економічній ситуації, правильно розраховувати свої витрати, а також здійснювати вигідні обмінні операції.

Основна Частина

Архітектура розробленого Telegram-бота побудована на основі фреймворку `aiogram` для створення ботів мовою Python. Взаємодія з Telegram Bot API здійснюється через об'єкти класів **Bot** і **Dispatcher**, які забезпечують отримання та обробку повідомлень від користувачів.

Ключовим компонентом системи є механізм обробників подій, реалізований через декоратори бібліотеки `aiogram`. Для кожного типу вхідних повідомлень (команда, текст тощо) визначені відповідні обробники, які викликаються асинхронно при надходженні даних.

Основні функції бота включають:

1. **Обробка команди `/start`**: Ця команда ініціалізує бота та виводить вітальне повідомлення з інструкціями. Користувачу одразу пропонується головне меню з клавіатурою.

```
@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(row_width=2)
    btn1 = types.KeyboardButton('Погода')
    btn2 = types.KeyboardButton('Валюта')
    btn3 = types.KeyboardButton('Конвертація валют')
    markup.add(btn1, btn2, btn3)
    bot.send_message(message.chat.id, 'Привіт, обери
що ти хочеш дізнатися:', reply_markup=markup)
```

Користувач отримує повідомлення з меню, яке містить кнопки "Погода", "Валюта" та "Конвертація валют".

2. **Обробка команди "Погода"**: Бот запитує у користувача назву міста та надає актуальну інформацію про погоду в цьому місті, використовуючи API `OpenWeatherMap`.

```
@bot.message_handler(content_types=['text'])
def handle_message(message):
    text = message.text.strip().lower()
    if text == 'погода':
        bot.send_message(message.chat.id, 'Введіть
назву міста для отримання погоди:')
        bot.register_next_step_handler(message,
get_weather)

def get_weather(message):
    city = message.text.strip()
    weather = get_weather_data(city)
    bot.send_message(message.chat.id, weather)

def get_weather_data(city):
    # Запит до API OpenWeatherMap
```

```

url = f"http://api.openweathermap.org/data/2.5/
weather?q={city}&appid={WEATHER_API}"
response = requests.get(url)
data = response.json()
if data['cod'] == 200:
    main = data['main']
    weather_desc = data['weather'][0]['description']
    temp = main['temp'] - 273.15 # Конвертація з
Кельвіна в Цельсій
    return f"Погода в {city}:\nТемпература:
{temp:.2f}°C\nОпис: {weather_desc}"
else:
    return "Місто не знайдено, будь ласка,
перевірте введену назву."

```

3. Обробка команди "Валюта": Бот запитує у користувача назву міста та надає інформацію про валюту країни, в якій знаходиться це місто, використовуючи API Rest Countries.

```

@bot.message_handler(content_types=['text'])
def handle_message(message):
    text = message.text.strip().lower()
    if text == 'валюта':
        bot.send_message(message.chat.id, 'Введіть
назву міста для отримання інформації про валюту:')
        bot.register_next_step_handler(message,
get_currency)

def get_currency(message):
    city = message.text.strip()
    currency = get_currency_data(city)
    bot.send_message(message.chat.id, currency)

def get_currency_data(city):
    # Запит до API Rest Countries
    url = f"https://restcountries.com/v3.1/capital/{city}"
    response = requests.get(url)
    data = response.json()
    if data:
        currency_info = data[0]['currencies']
        currency_name = list(currency_info.keys())[0]
        currency_symbol = currency_info[currency_name]['symbol']

```

```

        return f"Валюта країни, де знаходиться
{city}:\nНазва:                {currency_name}\nСимвол:
{currency_symbol}"
    else:
        return "Місто не знайдено, будь ласка,
перевірте введене назву."

```

4. Обробка команди "Конвертація валют": Бот запитує у користувача суму та надає можливість вибору валютної пари для конвертації.

```

@bot.message_handler(content_types=['text'])
def handle_message(message):
    text = message.text.strip().lower()
    if text == 'конвертація валют':
        bot.send_message(message.chat.id, 'Введіть
суму для конвертації:')
        bot.register_next_step_handler(message,
get_amount)

def get_amount(message):
    global amount
    amount = float(message.text.strip())
    markup = types.ReplyKeyboardMarkup(row_width=2)
    btn1 = types.KeyboardButton('USD/EUR')
    btn2 = types.KeyboardButton('EUR/USD')
    btn3 = types.KeyboardButton('USD/GBP')
    btn4 = types.KeyboardButton('Друге значення')
    markup.add(btn1, btn2, btn3, btn4)
    bot.send_message(message.chat.id, 'Виберіть
валютну пару для конвертації:', reply_markup=markup)

@bot.message_handler(content_types=['text'])
def handle_conversion(message):
    text = message.text.strip().lower()
    if text in ['usd/eur', 'eur/usd', 'usd/gbp']:
        from_currency, to_currency = text.split('/')
        converted_amount = convert_currency(amount,
from_currency, to_currency)
        bot.send_message(message.chat.id,
f"Конвертована сума: {converted_amount:.2f}
{to_currency}")
    elif text == 'друге значення':
        bot.send_message(message.chat.id, 'Введіть
пару кодів валют (наприклад, USD/EUR):')
        bot.register_next_step_handler(message,
custom_conversion)

```

```

def custom_conversion(message):
    from_currency, to_currency =
message.text.strip().upper().split('/')
    converted_amount = convert_currency(amount,
from_currency, to_currency)
    bot.send_message(message.chat.id, f"Конвертована
сума: {converted_amount:.2f} {to_currency}")

def convert_currency(amount, from_currency,
to_currency):
    # Використання бібліотеки CurrencyConverter для
конвертації валют
    converted_amount = currency.convert(amount,
from_currency, to_currency)
    return converted_amount

```

Для реалізації користувацького інтерфейсу використовуються спеціальні клавіатури, створені за допомогою класу **ReplyKeyboardMarkup**. Головне меню містить кнопки для переходу до режимів отримання погоди, валютних курсів та конвертації валют.

Висновок

Розробка Telegram-бота є актуальною задачею в сучасному глобалізованому світі, де швидкий та точний доступ до інформації відіграє важливу роль. Створений бот забезпечує користувачам зручний інструмент для отримання актуальних даних про погоду, валютні курси та здійснення конвертації валют, використовуючи сучасні технології машинного перекладу та обробки природної мови.

У процесі роботи було реалізовано ключові функції бота, включаючи отримання даних про погоду, валютні курси та конвертацію валют. Використання фреймворку aiogram дозволило створити ефективний та стабільний бот з інтуїтивно зрозумілим інтерфейсом. Основними перевагами розробленого бота є його модульність, гнучкість та можливість подальшого розширення функціоналу. Застосування принципів об'єктно-орієнтованого програмування забезпечило зручність супроводу та розвитку проекту.

Список літератури

1. Googletrans: Free and Unlimited Google translate API for Python [Електронний ресурс]. Режим доступу: <https://py-googletrans.readthedocs.io/en/latest/>. Дата доступу: 15.04.2024
2. aiogram 3.7.0 documentation [Електронний ресурс]. Режим доступу: <https://docs.aiogram.dev/en/latest/>. Дата доступу: 15.04.2024
3. OpenWeatherMap API documentation [Електронний ресурс]. Режим доступу: <https://openweathermap.org/api>. Дата доступу: 15.04.2024
4. Rest Countries API documentation [Електронний ресурс]. Режим доступу: <https://restcountries.com>. Дата доступу: 15.04.2024

МЕТОДИ ЗНАХОДЖЕННЯ КЛЮЧОВИХ СЛОВОСПОЛУЧЕНЬ

Олена Шапошнікова¹, Андрій Білик²

¹*Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaep@gmail.com*

²*Харківський національний автомобільно-дорожній університет, Харків.*

В сучасному світі майже всі документи створюються в електронному вигляді та зберігаються на комп'ютері. Однак документів настільки багато, що виникає складність у здійсненні ефективного пошуку потрібного документу. Для спрощення цього процесу виникає необхідність в їхньому індексуванні. Проте розміри текстового об'єму можуть бути настільки великі, що ускладнюють завантаження до Azure [1] або до баз даних, для їх подальшого пошуку.

Для розв'язання даної задачі доступні різноманітні алгоритми, включаючи «Sрасу» та «RAKE-NLTK» [2]. Більшість існуючих алгоритмів, які використовуються у програмах, спрямовані на зменшення обсягу тексту з метою зберігання потрібної інформації на сервері (у базі даних) для зменшення кінцевого обсягу даних. Але ці алгоритми можуть бути не ефективними з точки зору швидкодії, особливо на слабких системах.

Наразі одним із найбільш перспективних методів для вирішення задачі скорочення обсягу тексту є алгоритм «RAKE» (Rapid Automatic Keyword Extraction) [3]. Цей метод базується на принципі аналізу частотності та поєднання слів у тексті, дозволяючи виділити найбільш важливі терміни без необхідності попереднього навчання моделі.

Система «RAKE» виконує наступні основні функції:

- розбиття введеного тексту на окремі слова або фрази (цей крок є важливим для подальшого аналізу слів);
- визначення можливих ключових слів шляхом аналізу шаблонів, що включає в себе виділення слів, які часто зустрічаються в тексті;
- надання балів кожному слову на основі його частоти входження складових слів;
- аналіз оцінок потенційних ключових слів, ранжування їх за популярністю та вибір найбільш важливих термінів або фраз у якості кінцевих вилучених ключових слів.

При пошуку надто великих файлів «RAKE» не завжди справляється з задачею стискання розміру тексту до розмірів, які дозволяють завантажити його до Azure. Тому було розроблено інший алгоритм, який ще ефективніше зменшує обсяг тексту – це «Frequently occurring». «Frequently occurring» працює дуже швидко і ефективно на слабких системах та дуже сильно зменшує об'єм тексту. Варто зауважити, що алгоритм «Frequently Occurring» не визначає ключові словосполучення, а замість цього запам'ятовує ключові слова та не видаляє символи з тексту. А це може негативно вплинути на подальший пошук цього файлу .

На рис. 1 зображено інтерфейс програми індексації файлів «Index File»[4], де для проведення дослідження застосовуються ці два алгоритми.

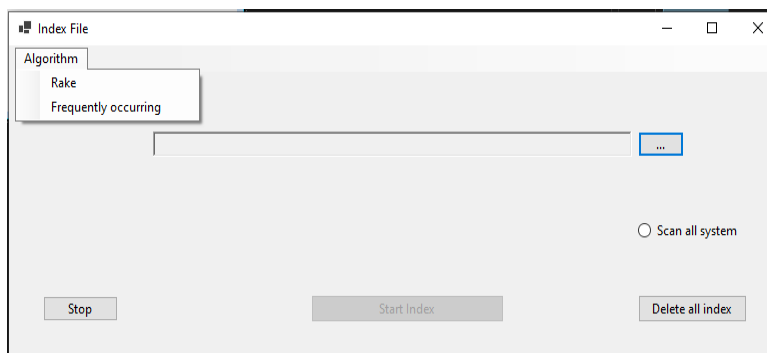


Рисунок 1 – Інтерфейс «Index File»

Для оцінки ефективності роботи алгоритмів «RAKE» та «Frequently Occurring» був здійснений порівняльний аналіз.

«Frequently occurring» виявляє часто повторювані слова, букви та символи, які зустрічаються більше 1-го разу. На рисунок 2 наведено результати роботи цього алгоритму.

KeyValuePair2	KeyValuePair2.key	KeyValuePair2.value	KeyValuePair2.Key	KeyValuePair2.Value
[міністерство, 1]	міністерство	1	міністерство	1
[освіти, 2]	освіти	2	освіти	2
[і, 17]	і	17	і	17
[науки, 1]	науки	1	науки	1
[україни харківський, 1]	україни харківський	1	україни харківський	1
[національний, 1]	національний	1	національний	1
[автомобільно, 1]	автомобільно	1	автомобільно	1
[дорожній, 1]	дорожній	1	дорожній	1
[університет	університет	1	університет	1
кафедра, 1]	кафедра		кафедра	
[комп'ютерних, 1]	комп'ютерних	1	комп'ютерних	1
[технологій, 2]	технологій	2	технологій	2
[мехатроніки	мехатроніки	1	мехатроніки	1
курсowa, 1]	курсowa		курсowa	
[робота1	робота1	1	робота1	1
з, 1]	з		з	
[дисципліни, 1]	дисципліни	1	дисципліни	1
[«програмування, 1]	«програмування	1	«програмування	1
[в, 39]	в	39	в	39
[ос, 1]	ос	1	ос	1
[android»	android»	1	android»	1
на, 1]	на		на	
[тему	тему	1	тему	1
«розробка, 1]	«розробка		«розробка	
[android, 190]	android	190	android	190
[додатку, 16]	додатку	16	додатку	16
[«читалка, 2]	«читалка	2	«читалка	2
[електронних, 7]	електронних	7	електронних	7
[книг»]	книг»	1	книг»	1
[студента, 1]	студента		студента	

Рисунок 2 – Робота алгоритму «Frequently occurring»

«RAKE» [5] спочатку розбиває текст на окремі слова або фрази, а потім створює список можливих ключових слів. Він надає бали кожному кандидату в ключові слова на основі його частоти та частоти складових його слів. На рисунку 3 представлені результати роботи алгоритму «RAKE», який визначає ключові словосполучення.

KeyValuePair	Key	Value	Key	Value
[public static final int database_version=1, 16.529505582137162]	public static final int database_version=1	16.529505582137162	public static final int database_version=1	16.529505582137162
[public static void additionsdb, 12.13090909090909]	public static void additionsdb	12.13090909090909	public static void additionsdb	12.13090909090909
[final int pageCount = pdfrenderer, 11.707827260458838]	final int pageCount = pdfrenderer	11.707827260458838	final int pageCount = pdfrenderer	11.707827260458838
[public static void updatepage, 11.63090909090909]	public static void updatepage	11.63090909090909	public static void updatepage	11.63090909090909
[public static int, 9.196172248803828]	public static int	9.196172248803828	public static int	9.196172248803828
[public void onupgrade, 9.13090909090909]	public void onupgrade	9.13090909090909	public void onupgrade	9.13090909090909
[public boolean ontouch, 9.09090909090909]	public boolean ontouch	9.09090909090909	public boolean ontouch	9.09090909090909
[integer primary key, 9]	integer primary key	9	integer primary key	9
[// closing streams, 8.75]	// closing streams	8.75	// closing streams	8.75
[public void onactivityresult, 8.13090909090909]	public void onactivityresult	8.13090909090909	public void onactivityresult	8.13090909090909
[public void deletedb, 8.13090909090909]	public void deletedb	8.13090909090909	public void deletedb	8.13090909090909
[public void deleteallfilesindirectory, 8.13090909090909]	public void deleteallfilesindirectory	8.13090909090909	public void deleteallfilesindirectory	8.13090909090909
[public void oncreate, 8.13090909090909]	public void oncreate	8.13090909090909	public void oncreate	8.13090909090909
[public void back, 8.13090909090909]	public void back	8.13090909090909	public void back	8.13090909090909
[public void onbackpressed, 8.13090909090909]	public void onbackpressed	8.13090909090909	public void onbackpressed	8.13090909090909
[protected void oncreate, 8.04]	protected void oncreate	8.04	protected void oncreate	8.04
[protected void onresume, 8.04]	protected void onresume	8.04	protected void onresume	8.04
[protected void ondestroy, 8.04]	protected void ondestroy	8.04	protected void ondestroy	8.04
[private void addcachefile, 8.04]	private void addcachefile	8.04	private void addcachefile	8.04
[public void onclick, 7.9880519480519485]	public void onclick	7.9880519480519485	public void onclick	7.9880519480519485
[separator + namebook, 7.978494623655914]	separator + namebook	7.978494623655914	separator + namebook	7.978494623655914
[public void additioninfodb, 7.797575757575758]	public void additioninfodb	7.797575757575758	public void additioninfodb	7.797575757575758
[// reading bytes, 7.75]	// reading bytes	7.75	// reading bytes	7.75
[// decoding bytes, 7.75]	// decoding bytes	7.75	// decoding bytes	7.75
[private void openfile, 7.706666666666667]	private void openfile	7.706666666666667	private void openfile	7.706666666666667
[protected void onpause, 7.706666666666667]	protected void onpause	7.706666666666667	protected void onpause	7.706666666666667
[private pdfrenderer pdfrenderer = null, 7.692307692307692]	private pdfrenderer pdfrenderer = null	7.692307692307692	private pdfrenderer pdfrenderer = null	7.692307692307692
[private appbarconfiguration appbarconfiguration, 7.666666666666668]	private appbarconfiguration appbarconfiguration	7.666666666666668	private appbarconfiguration appbarconfiguration	7.666666666666668
[// write info, 7.666666666666667]	// write info	7.666666666666667	// write info	7.666666666666667
[public void opendirbtnclick, 7.630909090909091]	public void opendirbtnclick	7.630909090909091	public void opendirbtnclick	7.630909090909091
[public void readdbandonclick, 7.630909090909091]	public void readdbandonclick	7.630909090909091	public void readdbandonclick	7.630909090909091
[private int cout = 0, 7.605263157894736]	private int cout = 0	7.605263157894736	private int cout = 0	7.605263157894736
[public void opendirf, 7.530909090909091]	public void opendirf	7.530909090909091	public void opendirf	7.530909090909091
[private int pos = 0, 7.332533885167464]	private int pos = 0	7.332533885167464	private int pos = 0	7.332533885167464
[public string extractnamefromuri, 7.285333333333335]	public string extractnamefromuri	7.285333333333335	public string extractnamefromuri	7.285333333333335
[pos + 1 +, 7.0175953079178885]	pos + 1 +	7.0175953079178885	pos + 1 +	7.0175953079178885

Рисунок 3 – Робота алгоритму «RAKE»

Таким чином було виконано порівняльний аналіз алгоритмів, «Frequently occurring» і «RAKE», результати якого наведені у табл. 1.

Таблиця 1 – Результати порівняльного аналізу алгоритмів «Frequently occurring» і «RAKE».

Критерій	RAKE	Frequently occurring
Швидкість	-	+
Виділення ключових слів	-	+
Виділення ключових словосполучень	+	-
Розмір тексту	-	+

В результаті проведеного дослідження було визначено, який з цих алгоритмів більше підходить для збереження сенсу тексту, а який для зменшення його розміру.

По результатам дослідження можна зробити висновки, що алгоритм «Frequently occurring» більше підходить для ефективного зменшення об'єму тексту та знаходження слів, які часто зустрічаються, при чому у рейтинг включаються не лише слова, а й інші знаки, які часто зустрічаються, але при пошуку файлу по словосполученням, може бути не ефективним.

«RAKE» особливо корисний у випадках, коли необхідно зберегти ключові словосполучення для подальшого пошуку файлів, де ці словосполучення зустрічаються. Його застосування включає можливість зменшити текст, для подальшої індексації, що полегшує і прискорює процес пошуку файлу.

Список використаних джерел

1. Що таке Azure index [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/ru-ru/azure/search/search-what-is-an-index>
2. Інші алгоритми [Електронний ресурс] – Режим доступу: <https://medium.com/international-school-of-ai-data-science/the-keyword-quest-exploring-automatic-keyword-extractors-db553c6ac229>
3. Що таке RAKE [Електронний ресурс] – Режим доступу: <https://ovchinnikov.cc/writing/rake/>
4. Реалізація програми [Електронний ресурс] – Режим доступу: <https://github.com/Haker53/azure-search-dotnet.git>

УДК 004.42.4

СТВОРЕННЯ ДОДАТКІВ НА WINDOWS FORMS: АРХІТЕКТУРА ТА ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ

Клячко М.М., Лебединський А.В.

Харківський національний автомобільно-дорожній університет, Харків

У сучасному світі, де Windows залишається однією з найпоширеніших операційних систем, розробка додатків для неї не втрачає своєї актуальності. Існує кілька різних платформ, які розробники можуть використовувати для створення програм для Windows. С# є відмінним вибором для створення програм для Windows завдяки тісній інтеграції з платформою .NET та інструментами розробки Microsoft. Ось деякі ключові аспекти використання С# для розробки програм під Windows:

- інтеграція з .NET Framework/.NET Core: С# щільно інтегрований з платформою .NET, що забезпечує широкі можливості розробки різноманітних додатків під Windows. Залежно від вимог проекту, ви можете вибрати .NET Framework [1] для традиційних настільних додатків або .NET Core/.NET 5+ для більш сучасних і кросплатформових додатків;

- Windows Presentation Foundation (WPF): WPF - це потужний фреймворк для створення настільних програм під Windows із використанням С#. WPF надає широкі можливості для створення інтерактивних інтерфейсів користувача за допомогою XAML (eXtensible Application Markup Language) і підтримує різні функції, такі як стилі, шаблони, анімації та багато іншого;

- Windows Forms (WinForms): Для створення класичних настільних програм під Windows також можна використовувати Windows Forms. Це більш

старий, але все ще широко використовуваний фреймворк, який надає простий і швидкий спосіб розробки програм із використанням C#;

- зручна інтеграція з Visual Studio: Visual Studio – основне середовище розробки для C#, і вона має багатий набір інструментів для створення програм під Windows. Вона пропонує широкі можливості для налагодження, дизайну інтерфейсу, створення звітів та багато іншого;

- доступ до бібліотек Windows API: C# дозволяє легко взаємодіяти з різними компонентами операційної системи Windows, такими як реєстр, файлова система, служби Windows та багато іншого завдяки доступу до багатого набору бібліотек Windows API [2];

- обробка подій та асинхронне програмування: C# має зручні засоби для обробки подій та асинхронного програмування, що робить розробку інтерактивних і чуйних додатків для Windows більш простою та ефективною.

Плюси та мінуси використання. Використання Windows Forms у C# забезпечує простоту та зручність у розробці настільних програм для Windows.

Плюси:

- простота: Створення інтерфейсу користувача за допомогою WinForms відбувається швидко і легко завдяки візуальному редактору і простоті використання компонентів;

- швидка розробка: Розробка програм на Windows Forms займає менше часу, ніж при використанні деяких інших технологій, завдяки зручному інструментарію та простоті мови програмування C#;

- інтеграція з .NET: WinForms без проблем інтегрується з іншими технологіями та бібліотеками до .NET Framework, що забезпечує доступ до широкого спектру можливостей для розробки;

- широке поширення: Використання WinForms дозволяє створювати програми, які можуть бути запущені на широкому діапазоні пристроїв під керуванням операційної системи Windows [3, 4].

Мінуси:

- обмежені можливості інтерфейсу: WinForms пропонує базові засоби для створення інтерфейсу, що може обмежувати можливості в галузі візуалізації та досвіду користувача, особливо в порівнянні з більш сучасними технологіями, такими як WPF.

- не підходить для мобільних пристроїв: Програми WinForms призначені в основному для настільних комп'ютерів під керуванням Windows і не можуть бути легко адаптовані для мобільних пристроїв або інших платформ.

- старіння: З появою нових технологій, таких як WPF та UWP, WinForms може розглядатися як застаріла технологія, хоча й залишається затребуваною для розробки певного типу програм.

Архітектура Windows Forms. **Форми (Forms):** Основний елемент інтерфейсу в WinForms. Форми представляють вікна додатку, на яких можуть розміщуватися інші елементи інтерфейсу, такі як кнопки, тексти, таблиці тощо. **Контролі (Controls):** Елементи інтерфейсу, що розміщуються на формах, такі як кнопки, тексти, полоси прокрутки, таблиці і т. д. Контролі можуть взаємодіяти з користувачем і виконувати різні функції. **Події (Events):** WinForms викорис-

товує модель подій для обробки взаємодії користувача з додатком. Контроли генерують події при взаємодії з користувачем (наприклад, натискання кнопки), і програміст може прив'язувати обробники подій для реагування на ці події. Модель розділення (Separation of Concerns): Хороша практика при розробці додатків полягає в розділенні логіки програми на різні шари або компоненти. Наприклад, розділення між логікою додатку і його графічним інтерфейсом.

Функціональні можливості Windows Forms. Графічний інтерфейс користувача (GUI): WinForms надає широкий набір контролів для створення і налаштування графічного інтерфейсу, що відповідає потребам додатку. Можливості макетування (Layout Capabilities): Ви можете використовувати різні менеджери макетування для організації розташування контролів на формі, наприклад, FlowLayoutPanel або TableLayoutPanel. Валідація введених даних (Data Validation): WinForms дозволяє встановлювати правила для валідації даних, що вводяться користувачем в тексти, наприклад, обов'язкові поля або правильний формат даних. Можливості друку (Printing Capabilities): Додатки, розроблені на WinForms, можуть використовувати можливості друку для надання звітів або іншої інформації у друкованій формі. Мультимедійні можливості (Multimedia Capabilities): Ви можете інтегрувати зображення, звуки і відео в додаток за допомогою контролів, таких як PictureBox або Windows Media Player. Мережеві можливості (Networking Capabilities): WinForms дозволяє взаємодіяти з мережевими ресурсами, використовуючи класи .NET для роботи з мережевими протоколами. Можливості доступу до баз даних (Database Access Capabilities): Ви можете підключати ваші додатки до різних типів баз даних, таких як SQL Server, MySQL або SQLite, використовуючи ADO.NET або Entity Framework.

Об'єднавши наявну інформацію, розглянемо застосування Windows Forms практично [5]. Була створена форма (рис. 1), яка включає реєстраційні поля для клієнтів.

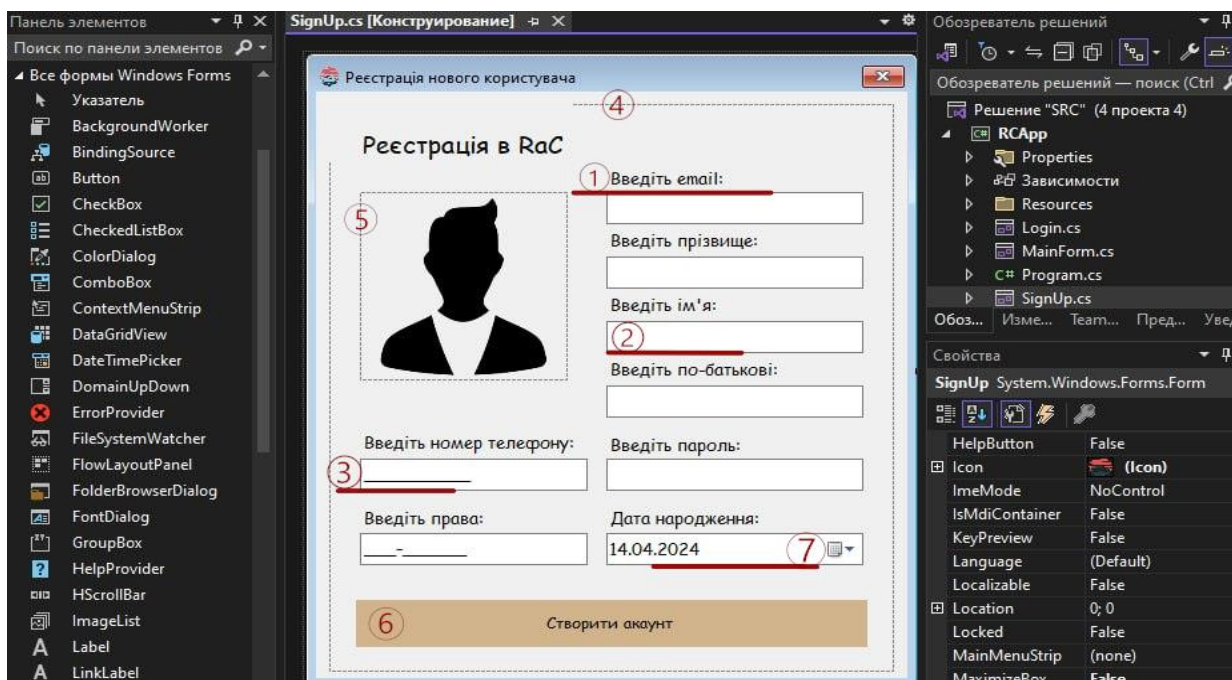


Рисунок 1 – Шаблон вікна реєстрації користувача

На формі були розміщені такі компоненти:

1. Label – компонент для написання тексту;
2. TextBox – компонент для введення тексту;
3. MaskedTextBox – поле (маска) для введення даних специфічного формату;
4. Panel – призначений для компонування об'єктів у вікні;
5. PictureBox – компонент для відображення зображень;
6. Button – кнопка;
7. DateTimePicker - календар, що розкривається за натисканням, у якому можна вибрати дату.

У висновку можемо сказати, що створення програм на Windows Forms представляє цікаву тему, яка охоплює як архітектурні аспекти, так і функціональні можливості цієї платформи.

Архітектура Windows Forms дозволяє розробникам швидко створювати графічні інтерфейси за допомогою компонентів, які легко розташовувати і керувати ними. Функціональні можливості включають широкий спектр можливостей для створення різноманітних додатків, від простих програм з однією формою до складних програм з багатьма вікнами, діалоговими вікнами і вбудованими елементами управління.

Список використаних джерел

1. Ефективний пошук в Інтернеті. *Ділова розвідка в Інтернеті*. URL: <http://isearch.kiev.ua/uk/news/programs/o-appl/1593-create-your-first-windows-forms-application-in-c-for-example-the-game-qtic-tac-toeq> (дата звернення: 10.04.2024).
2. Troelsen A., Japikse P. Pro C# 10 With .NET 6: Foundational Principles and Practices in Programming. Apress L. P., 2022.
3. C# Guide – .NET managed language. *Microsoft Learn: Build skills that open doors in your career*. URL: <https://learn.microsoft.com/uk-ua/dotnet/csharp/> (date of access: 10.04.2024).
4. Середовище програмування сі шарп та його переваги. *FoxmindEd*. URL: <https://foxminded.ua/seredovyshche-prohramuvannia-si-sharp/> (дата звернення: 10.04.2024).
5. C# та Windows Forms. Перший додаток. URL: <https://itproger.com/course/csharp-app> (дата звернення: 10.04.2024).

РЕКОМЕНДАЦІЙНІ СИСТЕМИ ТА ЇХ МОДЕЛІ

Лантєв Дмитро

Харківський національний автомобільно-дорожній університет

Рекомендаційні системи – це інформаційні системи, які надають користувачам персоналізовані рекомендації щодо товарів, послуг, вмісту або інших об'єктів на основі їхніх інтересів, поведінки або характеристик. Типи рекомендаційних систем визначаються основними методологіями та алгоритмами, використовуваними для генерації рекомендацій.

Основні типи рекомендаційних систем включають: фільтрація на основі контенту; колаборативна фільтрація; гібридні рекомендаційні системи; системи на основі знань.

Фільтрація на основі контенту є одним з типів рекомендаційних систем, в якому рекомендації генеруються на основі аналізу характеристик самого контенту (товарів, статей, фільмів і т.д.), а не залежностей між користувачами. Основна ідея полягає в тому, що подібний контент має високу ймовірність сподобатися користувачу, який проявив інтерес до певного елемента.

Колаборативна фільтрація є одним з типів рекомендаційних систем, в якому рекомендації генеруються на основі аналізу взаємодії користувачів з об'єктами. Основна ідея полягає в тому, що якщо два або більше користувачів мають схожі вподобання або взаємодіють з подібними об'єктами, то ймовірність того, що їм сподобається інший об'єкт, також висока. Гібридні рекомендаційні системи є комбінацією різних підходів та методів рекомендаційних систем з метою поліпшення точності та різноманітності рекомендацій. Вони поєднують у собі переваги різних типів систем для досягнення кращих результатів.

Системи на основі знань є одним з типів рекомендаційних систем і використовують знання про предметну область для генерації рекомендацій. Вони спираються на експертні знання, правила, моделі або онтології для розуміння вимог користувачів та об'єктів та надання рекомендацій.

Класичні моделі рекомендаційних систем є основними підходами та методами, які використовуються для створення рекомендаційних систем. Вони знаходять застосування у різних галузях, включаючи електронну комерцію, медіа та розваги, соціальні мережі, туризм та подорожі, освіту та навчання.

У сфері електронної комерції рекомендаційні системи допомагають пропонувати користувачам товари або послуги, які відповідають їхнім інтересам та попереднім покупкам.

У медіа та розважальній сфері рекомендаційні системи допомагають користувачам знайти відео, музику, фільми або книги, які відповідають їхнім смакам і інтересам. Вони аналізують вподобання користувачів, їхню історію переглядів або прослуховувань, і рекомендують відповідний контент.

Соціальні мережі використовують рекомендаційні системи для пропозицій нових друзів, груп або контенту на основі інтересів та зв'язків користу-

вачів. Алгоритми аналізують взаємодії, вподобання та активність користувачів, щоб знайти схожість та рекомендувати відповідний контент.

Також, рекомендаційні системи застосовуються у сфері освіти та навчання, допомагаючи студентам знаходити підходящі курси, матеріали та ресурси для навчання згідно їхніх академічних інтересів та потреб.

Класичні моделі рекомендаційних систем є важливими інструментами для персоналізації рекомендацій користувачів у різних сферах. Вони дозволяють аналізувати дані користувачів та об'єктів, враховувати їхні вподобання та створювати індивідуальні рекомендації, що покращують їхні досвіди та задоволення.

Література:

1. XLNet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems / Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. 2019. (pp. 5754-5764).

РОЗРОБЛЕННЯ ІС СОЦІАЛЬНОЇ ВЗАЄМОДІЇ. МОДУЛЬ «СОЦІАЛЬНА АДАПТАЦІЯ»

Олена ШАПОШНІКОВА¹, Іван СЕЩЕНКО²

¹Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaep@gmail.com

²Харківський національний автомобільно-дорожній університет, Харків.

У світі швидких змін та постійного руху суспільства, соціальна взаємодія стає важливішою ніж будь-коли раніше. Зокрема, для молодих людей, які шукають спільноту з подібними інтересами та хобі, розвиток інформаційних систем може стати вирішальним. У цьому контексті розробка інформаційної системи (ІС) для соціальної взаємодії з фокусом на важливу складову, «Соціальна адаптація», набуває великого значення.

В Україні та у світі загалом, соціальна мобільність в результаті конфліктів, економічних або особистих обставин часто змушує людей змінювати місце проживання. Такі зміни можуть бути важкими і створювати перешкоди для створення нового соціального кола. Інформаційні системи, спрямовані на полегшення соціальної взаємодії, стають важливим інструментом для адаптації та знаходження нових знайомств у нових місцях. За даними Мінреінтеграції в Україні на середину 2023 року вже налічувалось близько 5 мільйонів внутрішніх переселенців [1]. Значна частина людей, покинувши свої домівки, може більше не повернутись до старого життя, тому важливо надати їм можливість адаптуватися у новому місті.

В рамках вирішення цієї задачі пропонується розширити платформу соціальної взаємодії, яка була орієнтована на здійснення волонтерської допомоги унікальним модулем «Соціальна адаптація», який надасть можливість користувачам не тільки створювати власні події та заходи, але й приєднуватися до існуючих, спільно вирішуючи проблеми, знаходячи інтереси та хобі з подібними людьми. Передбачається, що сервіси цього модуля, як і сервіси усієї платформи будуть безкоштовними. Це зробить його доступним для широкого кола користувачів.

В цій роботі на першому етапі пропонується розробити мінімально життєздатний продукт - MVP (minimum viable product). Тобто, це буде продукт із достатньою кількістю функцій, щоб залучити клієнтів і перевірити ідею продукту на ранніх етапах циклу розробки продукту. MVP може допомогти якомога швидше отримати відгуки користувачів, щоб вдосконалювати продукт. MVP дозволить зібрати максимальну кількість перевірених знань про користувачів із найменшими зусиллями, бо, зважаючи на обставини, в яких знаходиться переважна більшість громадян України, є потреба випустити продукт на ринок якомога швидше. MVP дасть можливість протестувати ідею з реальними користувачами, перш ніж починати повну розробку продукту. І, врешті, дізнатися, що резонує з цільовою аудиторією, а що ні. Тобто, MVP дозволить підтвердити ідею для продукту без створення продукту в цілому, допоможе мінімізувати час і ресурси [2].

Тому передбачається, що основні функції, які закладаються на етапі створення MVP продукту такі:

- Створення та перегляд подій на одному сайті. Користувачі можуть не лише створювати свої події, але й легко знаходити та приєднуватися до існуючих без необхідності переходу на інші ресурси. Це забезпечує зручність та ефективність взаємодії.

- Спільнота та пошук. Система дозволить користувачам швидко знаходити людей з подібними інтересами та задачами, працевлаштуванням, спільно створювати заходи, а також обговорювати їхні вадливі теми в спеціальних групах чи чат-кімнатах.

Цільовою аудиторією на етапі MVP модуля є молоді люди віком від 18 до 35 років, які шукають нові контакти, однодумців та можливості для відпочинку та розвитку своїх інтересів.

Також сервіс буде корисним для людей, які змінили місце проживання та шукають спільноту в новому місті або країні, тому що велика кількість людей змінили місце проживання у межах країни, але ж і багато молодих людей були вимушені виїхати у інші країни, де соціальна адаптація відбувається вже набагато важче. За даними [3], близько 6 мільйонів людей були вимушені покинути Україну, значна кількість з це молодь, якій потрібно надати можливість знайти тих, хто зможе підтримати та зрозуміти їх у цей складний час, дозволити не втратити зв'язок з рідною країною та створити психологічне підґрунтя для повернення додому після закінчення воєнних дій.

Дослідження ринку показали, що наразі існують схожі платформи, які надають можливість створення подій та знаходження знайомств. Запропонована інформаційна система відрізняється тим, що вона безкоштовна. Тому у порівнянні з платними конкурентами, цей сервіс стає більш доступним та привабливим для широкої аудиторії.

Ця платформа забезпечує зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам швидко та ефективно знаходити та приєднуватися до подій. Вона має інтегрованість функцій – дозволяє як створювати, так і приєднуватися до подій в межах одного модуля. Це робить цей сервіс більш привабливим для користувачів, оскільки зменшується час та зусилля, необхідні для організації вирішення своєї задачі.

Адже інші системи, які, наприклад, надають можливість створювати візитки заходів, надають лише шаблонний сайт, а не платформу для розміщення своїх анонсів.

Розробка інформаційної системи соціальної взаємодії з модулем соціальної адаптації виявляється кроком вперед у розвитку та підтримці соціальної активності та мобільності.

Ця платформа пропонує зручні та безкоштовні інструменти для створення та приєднання до подій, сприяючи створенню нових соціальних зв'язків та підтримці вже існуючих. У світі, що постійно змінюється, інноваційні інформаційні системи стають ключовим елементом для підтримки та розвитку соціальної взаємодії.

Список використаних джерел

1. Суспільне новини. [Електронний ресурс] / Режим доступу: <https://suspilne.media/631564-v-ukraini-viplati-vpo-somisaca-otrimuut-26-mln-ludej-veresuk/#:~:text=В%20Україні%20майже%205%20мільйонів%20ВПО>
2. Мінімум життєздатний продукт (MVP). [Електронний ресурс] / Режим доступу: <https://www.productplan.com/glossary/minimum-viable-product/>
3. Ukrinform. [Електронний ресурс] / Режим доступу: <https://www.ukrinform.ua/rubric-economy/3815798-povernutisa-ci-zalisitisa-aki-cinniki-vplivaut-na-risenna-ukrainskih-bizenciv.html#>

ДЕМІСТИФІКАЦІЯ ВІЗУАЛІЗАЦІЇ ЧАСОВИХ РЯДІВ: ВИКОРИСТАННЯ ТЕХНОЛОГІЇ REACT.JS У ПРОЕКТІ STORAGE.TIMELINE

Олена ШАПОШНІКОВА¹, Герман СТЕПАНОВ²

¹*Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaer@gmail.com*

²*Харківський національний автомобільно-дорожній університет, Харків.*

У сучасному світі аналізу даних та моніторингу, розуміння та взаємодія з часовими рядами стає ключовим фактором для досягнення успіху в цільовому напрямку. Дані часових рядів, за своєю природою, є величезними та складними, часто охоплюючи безліч точок даних на протязі неперервного часового спектру. Величезний обсяг даних може бути приголомшливим, ускладнюючи виявлення закономірностей, тенденцій та аномалій.

З цією метою, командою розробників було створено інноваційний інструмент – Storage.Timeline. Storage.Timeline – це спеціалізована база даних, OpenSource проект створений для зберігання даних часових рядів, вона має ключові особливості для технології цього напрямку, а саме: мобільність зберігання, продуктивність обробки, можливість легко розширити стек для зберігання.

За участю членів дослідницької групи «Vitche» була зроблена велика кількість роботи, в результаті якої було створено архітектуру, яка описує принципи обробки даних загалом, і на основі цих принципів виникла архітектура Storage.Timeline. Таким чином, ця база даних спеціально розроблена для ефективного зберігання та отримання часових даних, використовуючи надійну технологію.

Однак, ця база даних сама по собі не забезпечує повний потенціал аналізу даних. Звідси розуміємо, що «Демістифікація» означає розгортання або розкриття того, що раніше було незрозумілим. У цьому контексті це означає розкриття складності аналізу часових рядів та візуалізації даних, саме тут важливу роль відіграє інтерфейс користувача (UI), який дає змогу за допомогою простих у використанні та водночас зрозумілих для користувача компонентів візуалізувати великий обсяг складних даних, легко виконувати навігацію по базі, використовуючи схеми, та переглядати часові лінії необхідних напрямків і категорій.

Виходячи з цього, для створення веб-інтерфейсів, щоб створити всебічний та зручний інтерфейс для взаємодії з даними часових рядів було прийняте рішення про застосування потужного інструменту React.js.

Ефективний UI для аналізу часових рядів має включати ряд ключових функцій. По-перше, легка навігація по схемі сховища – це дозволяє користувачам швидко переміщатися між різними схемами, що моделюють цільову область з перспективи часових рядів. Далі інтерактивний вибір діапазону часу

дозволяє користувачам фокусуватися на конкретних інтервалах, а можливості збільшення та панорамування дозволяють заглибитися в деталі або з легкістю отримати ширший огляд.

Завдяки реалізації потокової передачі даних, у реальному часі, користувачі можуть бути впевнені, що їх візуалізації завжди оновлені. Крім того, сюжети даних, які налаштовуються дозволяють задовольняти різноманітні аналітичні потреби, а анотації та маркери подій допомагають відзначати значущі моменти в даних для подальшого аналізу.

Використовуючи React.js, є можливість створити бібліотеку віджетів та використовувати її у вигляді npm пакету [1]. Ці пакети можуть містити в собі функції, класи, компоненти, методи та інші ресурси, які можна імпортувати та використовувати в JavaScript-коді, що зробить розробку програмного забезпечення більш ефективною, оскільки це дозволить уникати повторного написання коду та швидко використовувати готові рішення, які вже були реалізовані іншими розробниками, тим самим забезпечити інтеграцію віджету Storage.Timeline до будь-якого веб-сайту, створеного за допомогою сумісного фреймворку. Це дає можливість легко відображати необхідні візуальні елементи, не підключений віджет, або підключені віджети сховища, схеми та часові лінії, в залежності від потреб користувача.

Інтерфейс користувача для Storage.Timeline не лише допомагає взаємодіяти з даними часових рядів, але і максимізує їх потенціал для розуміння та прийняття обґрунтованих рішень. З кожним днем розвитку технологій, інструменти, які ми використовуємо для аналізу даних, також повинні розвиватися, щоб надати користувачам змогу повною мірою використовувати потенціал аналізу часових рядів.

Список використаних джерел

1. Npm [Електронний ресурс] – Режим доступу: <https://www.npmjs.com/>
2. Storage.Timeline документація [Електронний ресурс] – Режим доступу: <https://github.com/vitche/documentation-storage-timeline>

РОЗРОБКА БОТА-ПЕРЕКЛАДАЧА

Шамельов Владислав Олегович, Пронін С.В.

Харківський національний автомобільно-дорожній університет

У сучасному глобалізованому світі знання іноземних мов відіграє все важливішу роль для ефективної комунікації та обміну інформацією між людьми з різних країн і культур. Однак часто виникають ситуації, коли прямий переклад тексту з однієї мови на іншу стає необхідністю. У цьому контексті програмні перекладачі набувають великого практичного значення, дозволяючи швидко та якісно здійснювати переклад різних текстів.

Активний розвиток месенджерів і чат-ботів відкриває нові перспективи для створення зручних і доступних сервісів для перекладу тексту в діалоговому режимі. Телеграм-боти, побудовані на базі популярного месенджера Telegram, дозволяють реалізувати функції перекладача у вигляді простого та інтуїтивно зрозумілого інтерфейсу з використанням сучасних технологій машинного перекладу.

У даній роботі розглядається процес розробки телеграм-бота мовою Python для здійснення перекладу тексту з української мови на англійську і навпаки. Основною метою було створення програмного рішення з зручним користувацьким інтерфейсом та широким функціоналом, що включає можливість задавати напрямок перекладу, очищувати історію діалогу та отримувати підказки щодо роботи з ботом.

Актуальність створення ботів-перекладачів зумовлена зростаючою потребою в оперативному перекладі текстів у різних сферах людської діяльності.

Наявність простого і доступного інструменту для миттєвого перекладу відкриває нові можливості для міжмовної комунікації, обміну знаннями та культурними цінностями між людьми з різних країн і регіонів.

Важливо зазначити, що розробка ефективного бота-перекладача вимагає інтеграції сучасних технологій обробки природної мови, машинного перекладу та створення інтелектуальних інтерфейсів. У даному проекті були використані передові бібліотеки та фреймворки Python, такі як aiogram для взаємодії з Telegram Bot API та googletrans для здійснення перекладу тексту.

В рамках роботи було проведено аналіз існуючих рішень у галузі чат-ботів та машинного перекладу. Вивчені основні підходи та архітектурні принципи при створенні ботів, а також застосовані найкращі практики об'єктно-орієнтованого програмування для забезпечення гнучкості, розширюваності та зручності супроводу розробленого програмного коду.

Таким чином, розробка бота-перекладача є актуальним завданням, вирішення якого дозволяє розширити можливості міжкультурної комунікації та обміну інформацією в сучасних реаліях глобалізації.

Архітектура бота-перекладача

Архітектура розробленого бота-перекладача побудована на основі фреймворка aiogram для створення Telegram-ботів мовою Python. Взаємодія з Telegram Bot API здійснюється через об'єкти класів Bot і Dispatcher, які забезпечують отримання та обробку повідомлень від користувачів.

Ключовим компонентом системи є механізм обробників подій, реалізований через декоратори бібліотеки aiogram. Для кожного типу вхідних повідомлень (команда, текст тощо) визначені відповідні обробники, які викликаються асинхронно при надходженні даних.

Одним з основних обробників є функція `translate_text`, відповідальна за переклад введеного користувачем тексту. При активації режиму перекладу встановлюється глобальна змінна `translate_mode = True`. Потім при введенні тексту викликається метод `translate` з бібліотеки `googletrans`, який здійснює машинний переклад з вказаними вихідною та цільовою мовами:

```
translated_text = translator.translate(text, src=source_lang, dest=target_lang).text
```

Перекладений текст відправляється користувачу у вигляді відповідного повідомлення бота.

Для реалізації користувацького інтерфейсу використовуються спеціальні клавіатури, створені за допомогою класу `ReplyKeyboardMarkup`. Головне меню містить кнопки для переходу в режим перекладу, виклику інструкцій та очищення історії чату. В режимі перекладу відображається додаткова клавіатура з кнопками для зміни напрямку перекладу та скасування операції.

Інші ключові функції бота включають:

Обробку команди `/start` для ініціалізації бота та виведення вітального повідомлення з інструкціями. Користувачу одразу пропонується головне меню з клавіатурою. Обробку команди "Режим перекладу", яка активує режим перекладу тексту, встановлюючи спеціальний прапорець `translate_mode`. Бот виводить відповідне повідомлення, а користувачу показується клавіатура для роботи в цьому режимі.

Функцію `clear_chat` для очищення історії діалогу з ботом. Вона послідовно видаляє всі повідомлення в чаті, починаючи з останнього, використовуючи метод `bot.delete_message`.

Обробку команди "Інструкція", яка виводить повне керівництво по використанню бота з описом всіх функцій та команд.

Функцію `reverse_translation` для зміни напрямку перекладу з української на англійську або навпаки прямо в режимі перекладу тексту.

Кнопку "Скасувати" на клавіатурі режиму перекладу для виходу з цього режиму та повернення до головного меню.

Цей функціонал забезпечує зручне управління ботом та доступ до всіх основних можливостей для перекладу тексту в рамках інтуїтивно зрозумілого діалогового інтерфейсу.

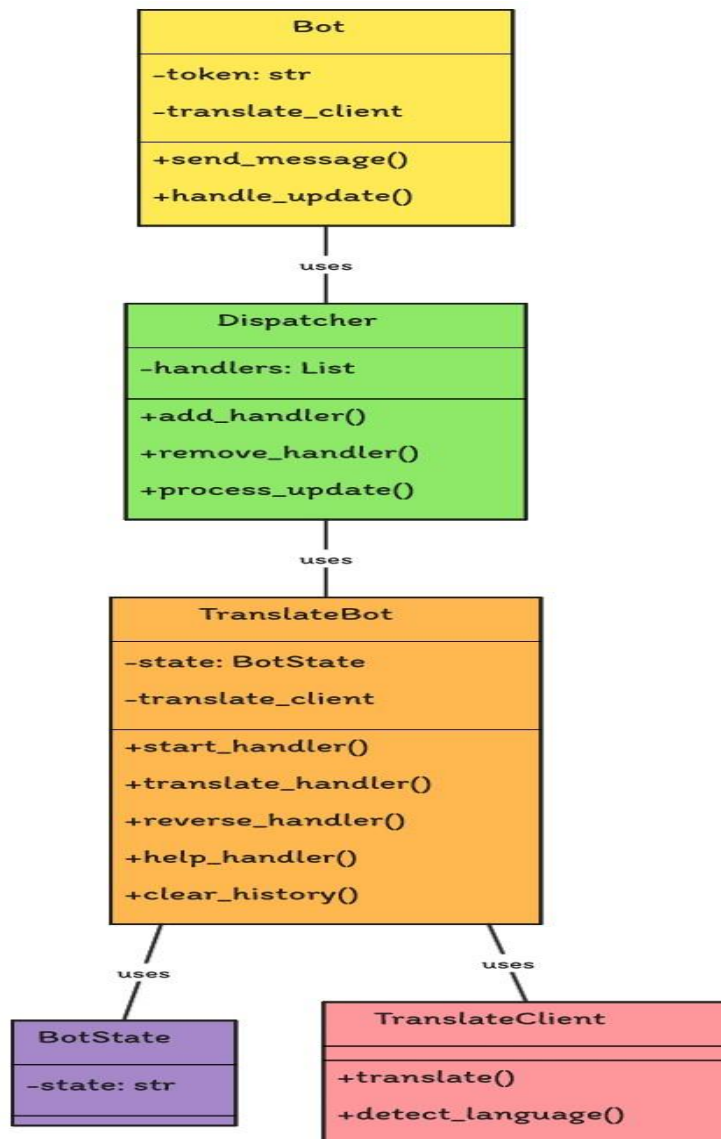


Рисунок 1.1 - Діаграма класів коду телеграм-бота

На діаграмі зображено основні класи та їхні взаємозв'язки:

Bot – головний клас, що інкапсулює взаємодію з Telegram Bot API.

Dispatcher – компонент, відповідальний за маршрутизацію вхідних оновлень до відповідних обробників.

TranslateBot – клас, що реалізує основну логіку перекладацького бота.

BotState – клас, що відстежує поточний стан бота.

TranslateClient – клас, що інкапсулює функції перекладу тексту.

Архітектура забезпечує модульність, розширюваність та легкість підтримки коду завдяки чіткому розподілу відповідальностей між класами.

Висновок

Розробка телеграм-бота перекладача є актуальною задачею в сучасному глобалізованому світі, де швидкий та точний переклад текстів відіграє важливу роль у міжкультурній комунікації та обміні інформацією. Створений бот

забезпечує користувачам зручний інструмент для перекладу текстів з української мови на англійську та навпаки, використовуючи сучасні технології машинного перекладу та обробки природної мови.

У процесі роботи було реалізовано ключові функції бота, включаючи активацію режиму перекладу, очищення історії чату, зміну напрямку перекладу, а також надання користувачам детальних інструкцій з використання бота. Використання фреймворку aiogram дозволило створити ефективний та стабільний бот з інтуїтивно зрозумілим інтерфейсом.

Основними перевагами розробленого бота є його модульність, гнучкість та можливість подальшого розширення функціоналу. Застосування принципів об'єктно-орієнтованого програмування забезпечило зручність супроводу та розвитку проекту.

Результати цієї роботи можуть бути корисними для студентів, викладачів, туристів, бізнесменів та всіх, хто потребує швидкого та якісного перекладу текстів у своєму повсякденному житті та професійній діяльності. Розробка бота-перекладача демонструє можливості сучасних технологій та відкриває нові перспективи для подальших досліджень і впроваджень у сфері автоматизованого перекладу текстів.

Список літератури

1. Googletrans: Free and Unlimited Google translate API for Python – Googletrans 3.0.0 documentation. Googletrans: Free and Unlimited Google translate API for Python – Googletrans 3.0.0 documentation. URL: <https://py-googletrans.readthedocs.io/en/latest/>.
2. aiogram 3.7.0 documentation. aiogram 3.7.0 documentation. URL: <https://docs.aiogram.dev/en/latest/>.
3. Карманов І. За кордоном Hello World: повний гід з розробки Telegram ботів за допомогою Python та Aiogram 3. Частина 1. Хабр. URL: <https://habr.com/ru/articles/732136/>.

РОЗРОБКА ANDROID-ДОДАТКІВ НА JAVA ТА KOTLIN: ПЕРЕВАГИ ТА НЕДОЛІКИ

Щербініна Ю. В., Лебединський А. В.

Харківський національний автомобільно-дорожній університет, Харків

Розробка мобільних додатків стала невід'ємною частиною сучасного програмного забезпечення. Android, як одна з найпопулярніших мобільних платформ, надає розробникам широкий спектр інструментів та можливостей для створення інноваційних та корисних додатків.

Android, найпоширеніша в світі операційна система для мобільних пристроїв, має безліч джерел підтвердження цього факту. iOS займає друге місце, проте різниця в частці ринку значно велика. За статистикою, Android володіє часткою ринку у 72,2%, тоді як iOS – у 26,99% [1].

Java та Kotlin – це дві мови програмування, які широко використовуються для розробки Android-додатків.

Java — офіційна мова програмування на Android. Вона була першою і залишається найпоширенішою, навіть після появи Kotlin.

Kotlin — це сучасна, багатоцільова мова програмування, яка зародилася в JetBrains і швидко набула широкої популярності серед розробників.

У цій роботі буде проведено порівняльний аналіз Java та Kotlin з точки зору їхніх переваг та недоліків у контексті розробки Android-додатків.

Java для розробки Android

Переваги:

- зріла та добре підтримувана мова з багаторічною історією;
- велика спільнота розробників та доступність навчальних ресурсів [2];
- висока продуктивність та ефективність коду;
- глибока інтеграція з платформою Android та базовими API;
- постійні оновлення для спрощення роботи [3] ;
- наявність великої кількості бібліотек та готових віджетів для спрощення роботи (рис. 1, рис. 2).
- висока безпека за рахунок роботи віртуальної машини.
- підтримка Google.
- розширені можливості при використанні потоків, анотацій (рис. 3) і т.д. Використання анотацій в Java має кілька переваг [4]:
- анотації дозволяють додавати додаткові метадані до коду.
- підтримка інструментів.
- підвищення читабельності коду.

Недоліки:

- багатослівний синтаксис, що може призвести до довшого та складного коду;

- потребує велику кількість ручної роботи, що може призвести до більшої кількості помилок [5];
- жорсткість мови, яка може ускладнювати деякі задачі, такі як функціональне програмування.

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/contentRecyclerView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/moduleName" />

```

Рисунок 1 – Використання віджету для прокручуваного списку RecyclerView

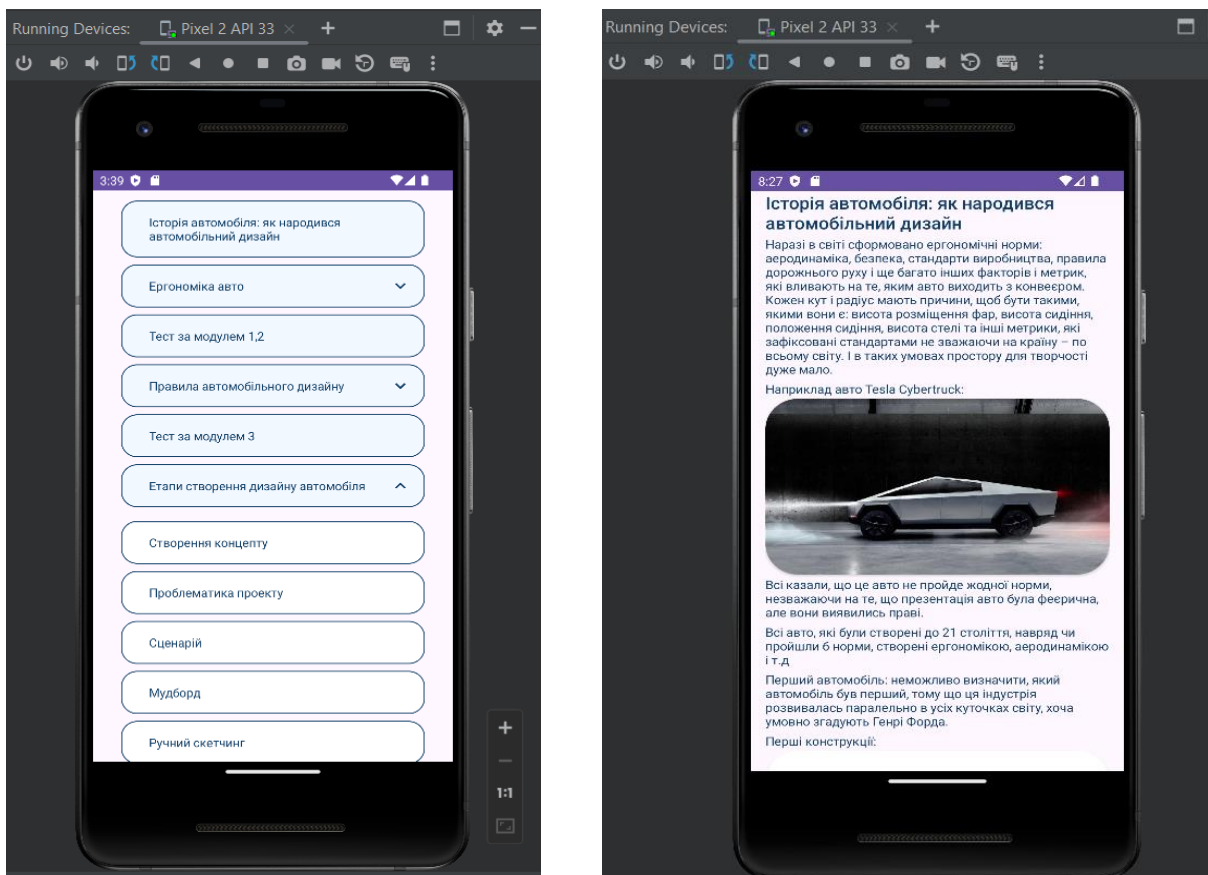


Рисунок 2 – Відображення віджету RecyclerView за допомогою емулятора всередині додатку Android Studio

```
CalendarActivity.java x activity_auth.xml x BaseActivity.java x ChatsActivity.java x ModuleContentActivity.java x ModulesActivity.java x SubModulesListAdapter.java x
28
29 @Override
30 protected void onCreate(Bundle savedInstanceState) {
31     super.onCreate(savedInstanceState);
32     setContentView(R.layout.activity_calendar);
33     attachBottomNav(R.id.calendar);
34
35     Calendar calendar = Calendar.getInstance();
36     int dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
37     int month = calendar.get(Calendar.MONTH);
38     int year = calendar.get(Calendar.YEAR);
39     int dayOfWeek = calendar.get(Calendar.DAY_OF_WEEK);
40     // Массив назв місяців
41     String[] monthNames = {"січня", "лютого", "березня", "квітня", "травня", "червня", "липня", "серпня", "вересня", "жовтня",
42
43     // Формування рядка з потрібним форматом дати
44     String formattedDate = String.format("%d %s %d", dayOfMonth, monthNames[month], year);
45
46     // Вивід інформації про поточну дату у встановленому форматі
47     System.out.println(formattedDate);
48     Log.d("TAG", "msg: " + formattedDate);
49
50
51     // Створення об'єкту SimpleDateFormat для форматування місяця
52     SimpleDateFormat dateFormat = new SimpleDateFormat("EEEE, d MMMM", Locale.getDefault());
53     SimpleDateFormat dateFormatYear = new SimpleDateFormat("yyyy", Locale.getDefault());
```

Рисунок 3 – Використання анотації @Override у коді для кастомізації календаря

Kotlin для розробки Android

Під час розробки мови Kotlin програмісти ставили перед собою задачу створити мову таку, щоб вона була лаконічною, безпечною і сумісною з Java, що в результаті зробило її привабливою альтернативою Java для багатьох розробників [6].

Переваги:

- лаконічний та читабельний синтаксис, що робить код більш зрозумілим та простим у підтримці.
- наявність сучасних функцій, таких як розширення функцій, автоматичне визначення null (помилки NullPointerException), що допомагає підвищити продуктивність розробки [7].
- міжплатформність, що дозволяє використовувати код Kotlin для розробки не лише Android-додатків, але й веб-сайтів та back-end-сервісів [8].

Недоліки:

- молодша мова порівняно з Java, що може призвести до меншої екосистеми інструментів та бібліотек.
- не така звична для Android, як Java, тому може знадобитися більше часу, щоб навчитися з нею працювати.
- можливі проблеми з продуктивністю в деяких випадках, коли код Kotlin може бути менш ефективним, ніж код Java.

У якості висновку можна сказати, наприклад, щоб виконати одну й ту саму дію, в Kotlin потрібно на декілька відсотків менше коду, ніж в Java, а сам він буде набагато стилістично простіше, в класичній мові є досить такого, чого немає у більш сучасного Kotlin.

Таблиця 1 – Порівняльна характеристика мов

Критерій	Java	Kotlin
Зрілість	Зріла	Молодша
Спільнота	Велика	Зростаюча
Продуктивність	Висока	Може бути трохи нижчою
Синтаксис	Багатослівний	Лаконічний
Сучасні функції	Доступні багато	Відсутні деякі
Міжплатформність	Ні	Так
Навчання	Багато ресурсів	Менше ресурсів

Кожна мова має низку переваг та недоліків у використанні, але для себе я обираю мову програмування Java через її багатий функціонал, багаторічну підтримку розробниками та велику кількість документації, а також через свій досвід створення Android додатків на Java впродовж навчання в університеті.

Список використаних джерел

1. Java in App Development: Why Choose It for Mobile Applications | *EPAM Startups & SMBs. Software Development Services for Companies* | EPAM Startups & SMBs. URL: <https://startups.epam.com/blog/java-for-mobile-app-development> (date of access: 12.04.2024).
2. Частка ринку Android та iOS: статистика 2022 року. *Root-Nation.com*. URL: <https://root-nation.com/ua/news-ua/it-news-ua/ua-android-ios-statistika-2022/> (дата звернення: 12.04.2024).
3. SemperAnte. Java @Аннотації. Що це та як цим користуватися?. *JavaRush*. URL: <https://javarush.com/ua/groups/posts/uk.1896.java-annotac-jsho-ce-ta-jak-cim-koristuvatisja> (date of access: 12.04.2024).
4. Kotlin проти Java – різниця між ними. *Guru99*. URL: <https://www.guru99.com/uk/kotlin-vs-java-difference.html> (дата звернення: 12.04.2024).
5. Chiusano P., Vermeulen M., Vjarnason R. *Functional Programming in Kotlin*. Manning Publications Co. LLC, 2021.
6. Kotlin and Android | Android Developers. Android Developers. URL: <https://developer.android.com/kotlin> (date of access: 12.04.2024).
7. Reason to use Java instead of Kotlin for Android development in 2023. URL: https://www.reddit.com/r/androiddev/comments/10nq5wf/is_there_still_a_reason_to_use_java_instead_of/?rdt=40534 (date of access: 12.04.2024).
8. Reason to use Java instead of Kotlin for Android development in 2023. URL: https://www.reddit.com/r/androiddev/comments/10nq5wf/is_there_still_a_reason_to_use_java_instead_of/?rdt=40534 (дата звернення: 12.04.2024).

**СУЧАСНІ РИЗИКИ У ТРАНСПОРТІ:
ВИКЛИКИ ТА ШЛЯХИ ПОДОЛАННЯ**

Бегунов Олександр

Харківський національний автомобільно-дорожній університет

Транспорт є життєво важливою складовою сучасного суспільства, але приводячи до збільшення загроз та ризиків у різних сферах. Особливо важливими є проблеми, пов'язані з кібербезпекою, надмірним навантаженням та інфраструктурою, забезпечуючи рух товарів, послуг та людей. Проте разом з розвитком технологій та зростанням обсягів перевезень з'являються нові виклики та ризики, які можуть мати вплив на безпеку та стабільність транспортного сектору. У цій статті ми розглянемо найбільш актуальні сучасні ризики у транспорті та можливі шляхи їх подолання.

Кібербезпека

Кібербезпека на транспорті-це безпека персональних даних, якими користувачі обмінюються з транспортним обладнанням та агрегаторами транспортних даних.

Сьогодні розумні системи оточують нас повсюди. Вони контролюють автомобільні та залізничні дороги через платформи спостереження, відстежують та запобігають затори на дорогах як навігатори, відповідають за безпеку пасажирів і водіїв транспортних засобах.

Технічно можна уявити ситуацію, коли двигун або системи керування автомобіля, що їде по автостраді, вмикаються за допомогою дистанційного керування. Теоретично, хакери могли б керувати автомобілем віддалено від водія. Однак хвилюватися не варто. Це тому, що не всі транспортні засоби здатні це робити.

Оскільки транспорт стає все більш цифровим, загроза кібератак також зростає. Хакери можуть атакувати системи управління транспортною мережею, технічне обладнання в транспортних засобах, система онлайн-продажу квитків транспортні сервіси. Це може призвести до серйозних аварій, перебоїв у русі та втрати конфіденційної інформації. Для подолання цієї загрози необхідно посилити захист кібербезпеки у всіх сферах транспортної інфраструктури. Це включає розробку сильнішого антивірусного програмного забезпечення, впровадження шифрування даних і регулярне навчання персоналу з питань кібербезпеки.

Забруднення та зміна клімату

Транспорт має значний вплив на повітря, земельні та водні ресурси, біорізноманіття, зміну клімату, екосистеми в цілому та здоров'я населення. Основними видами впливу транспорту на навколишнє середовище є викиди вихлопних газів відходи від транспорту (наприклад, викиди технологічних рідин, частинки шин, побутові відходи) електромагнітні коливання забруднення водних об'єктів через морський та річковий транспорт; руйнування природних ландшафтів, зменшення площ насаджень та сільськогосподарських угідь, деградація земель внаслідок будівництва об'єктів транспортної мережі; порушення водоносних горизонтів через великі насипи під час будівництва залізниць, автомобільних доріг та злітно-посадкових смуг; зменшення ареалів існування тварин (птахів навколо аеропортів, тварин внаслідок будівництва доріг); переміщення немісцевих видів диких тварин за допомогою транспортних засобів. Викиди в атмосферу від пересувних джерел забруднення мають найбільший вплив на навколишнє середовище.

На автомобільний транспорт припадає 90% викидів шкідливих речовин, у тому числі: 94% викидів оксиду азоту, 92% викидів оксиду вуглецю, 90% викидів сажі, 75% викидів метану та неметанових органічних сполук, 70% викидів діоксиду сірки та 62-65% викидів діоксиду азоту. До них відносяться наступні. Частка автомобільного транспорту у викидах парникових газів зростає з 40,2% у 1990 році до 84,5% у 2011 році, як очікується, продовжить зростати.

Надмірне навантаження на інфраструктуру

Надмірне навантаження на транспортну інфраструктуру є серйозною проблемою в сучасному світі. Швидке зростання населення, розвиток міст та економічна експансія чинять тиск на транспортні системи, спричиняючи затори, забруднення та інші негативні наслідки. Попередні дослідження підтвердили, що надмірне навантаження на транспортну інфраструктуру стає все більшою проблемою в багатьох країнах світу. Затори стали нормою у великих містах, що призводить до збільшення часу в дорозі та споживання палива, а також до підвищення рівня забруднення повітря. Затори спричинені низкою факторів, серед яких зростання населення, нерозвиненість систем громадського транспорту, надмірне використання приватних транспортних засобів та неадекватна інфраструктура. Наслідки надмірного навантаження на транспортну інфраструктуру включають затори, що призводять до збільшення часу та витрат палива. Забруднення навколишнього середовища через викиди від транспортних засобів, що стоять. Збільшення кількості дорожньо-транспортних пригод і травм.

Висновок

Сучасні ризики у транспорті вимагають комплексного підходу та спільних зусиль уряду, приватного сектору та громадськості. Шляхи подолання цих ризиків включають в себе збільшення інвестицій у кібербезпеку, захист

довкілля та модернізацію транспортної інфраструктури. Тільки завдяки спільним зусиллям можна забезпечити безпеку та стійкість транспортного сектору у майбутньому.

Література

1. Сталий розвиток для України [Електронний ресурс]. Режим доступу: <https://sd4ua.org/golovni-temi-stalogo-rozvitku/transport/>. Дата доступу: 15.04.2024
2. Що таке кібербезпека? [Електронний ресурс]. Режим доступу: <https://www.microsoft.com/uk-ua/security/business/security-101/what-is-cybersecurity>. Дата доступу: 15.04.2024

УДК 005.8

ВИКОРИСТАННЯ ГНУЧКИХ МЕТОДІВ У ПРОЄКТНІЙ ДІЯЛЬНОСТІ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Петренко Ю.А., Бугаєвський М.С.

Харківський національний автомобільно-дорожній університет, Харків

Анотація. Вивчено передумови, які призвели до застосування послідовних та гнучких методів розробки програмного забезпечення. Проаналізовано основні аспекти, переваги та недоліки використання гнучких методологій у проєктній діяльності при розробці програмного забезпечення.

Ключові слова: agile, scrum, kanban, waterfall, розробка програмного забезпечення, управління проєктами.

Людство стало розробляти програмне забезпечення відносно недавно - розробка великих програмних систем почалася всього близько 50 років тому. Цілком природно, що ранні підходи до розробки були мало формалізовані і представляли собою процес, який прийнято називати Code-and-Fix (кодування і виправлення).

При такому підході розробка програмного забезпечення починається безпосередньо з кодування (без попереднього планування, аналізу вимог і проєктування). Після цього знайдені в коді проблеми (дефекти, невідповідність вимогам тощо) виправляються шляхом внесення множинних змін в код. Тому після деякої кількості таких змін система стає заплутаною, її складно підтримувати і розширювати.

Згодом стало зрозуміло, що для створення великих стійких програмних систем потрібні більш продумані й формальні підходи. У пошуках вирішення проблеми увагу було звернуто на більш зрілі на той час галузі людської діяльності, пов'язані зі складним виробництвом - перш за все, на системотехніку (systems engineering), а також інші інженерні дисципліни, такі як проєктування і будівництво мостів, споруд тощо.

У результаті спроби використовувати перевірені в інших областях інженерні методи для розробки програмного забезпечення з'явилася нова дисципліна - програмна інженерія (software engineering), а як фактичні стандарти на довгі роки утвердилися так звані інженерні методології розробки програмного забезпечення. Ці методології також часто називають заснованими на плані (plan-driven), тому що в їх основі лежить припущення про те, що процес розробки програмного забезпечення є детермінованим інженерним процесом, який можна спланувати від початку і до кінця та виконати відповідно до плану, використовуючи формальні інженерні підходи.

В якості основного варіанту побудови життєвого циклу використовувався водоспадний життєвий цикл (waterfall), що передбачає одноразовий прохід по фазах аналізу вимог, проектування, кодування, тестування тощо. Таким чином, перший фундаментальний зсув у області методологій розробки представляв собою перехід від хаосу підходу code-and-fix до строго формалізованого підходу, прийнятого в інженерних методиках.

Навіть у 60-70-х роках минулого століття існували окремі проекти, в яких використовувались інші варіанти життєвого циклу (як різні модифікації водоспадного, так і ітераційні). Однак такі проекти були, скоріше, винятком із правил, і в рамках цього дослідження ми визначаємо, перш за все, основні й масові напрямки та тенденції.

Починаючи з найперших програмних проектів і по нинішній день, розробка програмного забезпечення була й залишається малопередбачуваною та далеко не завжди успішною справою.

Значний відсоток проектів зі створення програмного забезпечення, як і раніше, завершується з перевищенням бюджету, термінів, а створені в результаті програми часто не до кінця відповідають вимогам користувачів або приносять мало реальної користі бізнесу. Перераховані проблеми є основними проявами так званої кризи програмного забезпечення.

Незважаючи на значні інтелектуальні зусилля, витрачені на пошук способів подолання кризи, досі так і не знайдено універсальне рішення. Гнучкі методи – це сучасна відповідь програмній індустрії на питання, як все-таки треба виконувати проекти, щоб вони з більшою долею ймовірності завершувалися успіхом і приносили користь усім зацікавленим сторонам - і перш за все, замовнику і команді проекту.

Починаючи розробку корпоративного програмного забезпечення, важливо розуміти, з яким стеком технологій можна підійти до вирішення завдань та які основні показники програмного продукту важливі для замовника.

Вибір нового програмного забезпечення великим підприємством є досить повільним процесом, коли понад 50% роботи виконується ще на етапі аналізу початкових вимог. Це пов'язано з основними викликами сучасного світу - високим ступенем автоматизації і швидкою адаптацією процесів і систем до потреб клієнтів і ринку. Співробітникам, які керують процесом розробки програмного забезпечення та беруть у ньому участь, необхідно приділяти увагу

ефективності, використовувати відповідні технології, стилі і методи управління для розробки великих програм. Саме завдяки ним можна координувати роботу різних підрозділів, від якості і швидкості взаємодії між якими залежить швидкість розробки програмних компонентів та їх відповідність поставленим вимогам, яка і визначає їх цінність для бізнесу.

Звідси й виникає необхідність у наявності підходу, притримуючись якого кожен член команди буде розуміти свою роль, свої завдання у роботі над проєктом та критерії оцінки фінального результату.

Для досягнення цієї мети може використовуватися гнучка методологія розробки програмного забезпечення для управління проєктами. Вона спрямована на застосування ітеративних підходів розробки, динамічну постановку вимог та забезпечення їхньої реалізації за допомогою взаємодії між членами самоорганізованих робочих груп, до складу яких можуть входити спеціалісти різних напрямів розробки та підтримки ПЗ. Найбільш розповсюдженими з тих, що відносяться до класу гнучких є декілька методологій розробки ПЗ: екстремальне програмування, DSDM, Scrum, Kanban [1].

Ці методології використовуються як основа для ефективної організації роботи груп невеликого розміру, які виконують однорідну роботу. Ціллю, яку намагаються досягнути гнучкі методології є розподіл загального об'єму роботи, необхідного для виконання проєкту, на дрібні етапи / ітерації (тривалість кожного декілька тижнів), що мінімізує велику кількість ризиків, найбільшим з яких є невдоволеність клієнта результатом роботи через те, що вимоги змінилися з часу початку проєкту.

Кожна ітерація схожа на невеликий програмний проєкт та включає всі етапи, які необхідно пройти для отримання атомарного приросту за функціональністю: планування, аналіз вимог, проєктування, програмування, тестування та документування. Тож завдяки цьому підходу після завершення кожного етапу клієнт матиме нову версію продукту / програми. Команда обов'язково має виконувати перегляд та переоцінку пріоритетів після завершення кожного етапу.

Життєвий цикл продукту при використанні гнучкої методології розробки представлений на рис. 1.

В основу комунікативних аспектів Agile-методів покладено особисте спілкування. У більшості випадків команда, яка притримується даного підходу розташовується в одному приміщенні, інколи до цих команд входить замовник проєкту або його представник (визначає вимоги до продукту, це може бути менеджер проєкту, бізнес-аналітик або клієнт). Команда включає в себе тестувальників, дизайнерів інтерфейсу, технічних спеціалістів та менеджерів.

Робочий продукт – це основна метрика, за якою вимірюється ефективність таких методів. При використанні в команді спілкування обличчя до обличчя, гнучкі методи дозволяють зменшити об'єм письмової документації порівняно з іншими методиками. Однак, ці методи отримали певну долю

критики в свій бік через те, що деякі спеціалісти вважають, що використання робить команду менш дисциплінованою [2].

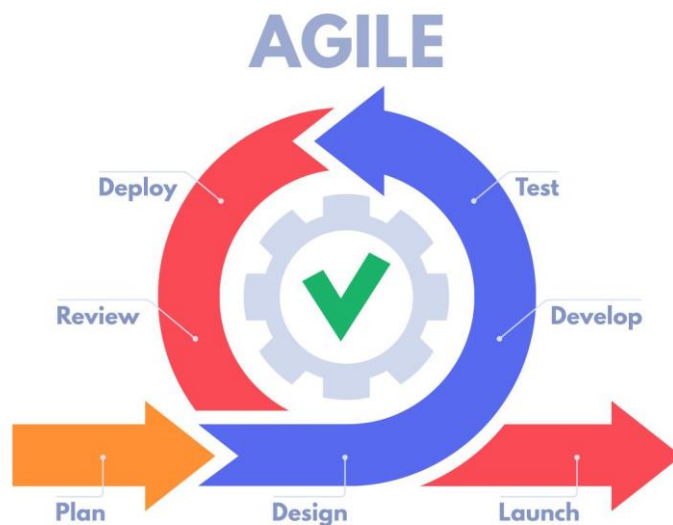


Рисунок 1 – Життєвий цикл продукту при використанні гнучкої методології розробки

Agile – це сімейство процесів розробки, а не єдиний підхід у розробці ПЗ, і основні його принципи описуються у Agile Manifesto. Agile Manifesto не містить практичних порад, але відображає чотири основні ідеї та дванадцять принципів Agile.

Основні ідеї:

1. Люди та взаємодія набагато важливіші за процеси та інструменти.
2. Продукт, що працює, важливіший за документацію.
3. Співпраця із замовником є важливішою за обговорені умови контракту.
4. Готовність до змін важливіше за слідування початковому плану.

Принципи:

1. Задовольнити клієнта ранньою та безперебійною поставкою цінного програмного забезпечення.
2. Адекватно сприймати зміни вимог навіть у кінці розробки, що може підвищити конкурентну спроможність продукту.
3. Постійна поставка робочого ПЗ, з постійною частотою (кожен місяць / тиждень).
4. Тісне та щоденне спілкування замовника з розробниками протягом усієї тривалості проєкту.
5. Проєктом займаються мотивовані особистості, які забезпечені необхідними умовами роботи, підтримкою та довірою.
6. Рекомендований метод передачі інформації – особиста розмова (face-to-face).

7. Робоче ПЗ – найкращий вимірювач прогресу.
8. Спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін.
9. Постійна увага сприяє покращенню технічної досконалості та зручному дизайну.
10. Простота – мистецтво не робити зайвої роботи.
11. Найкращі технічні вимоги, дизайн та архітектура створюються самоорганізованою командою.
12. Постійна адаптація до обставин, що змінюються. Команда зобов'язана систематично аналізувати можливі способи покращення ефективності та відповідно корегувати стиль своєї роботи [3].

До недоліків цього підходу можна віднести наступні важливі моменти.

Під час використання Agile-підходу дуже часто нехтують створенням плану дій, спрямованих на розвиток продукту на довгий період часу, та управлінням вимогами. Він надає можливість замовнику в будь-який момент, незважаючи на етап виконання, виставити нові вимоги, які можуть суперечити архітектурі продукту, який вже частково створено. Такий підхід інколи може призводити до катастрофічних наслідків: перероблення практично на кожній ітерації [4].

Вважають, що робота в Agile мотивує розробників вирішувати всі поставлені задачі достатньо простим та швидким способом. Таким чином, виконавці проекту можуть не звертати увагу на правильність написання коду з урахуванням вимог використаної в основі платформи. І вже після початку використання розробленого продукту можуть виникнути конфліктні ситуації за технічною частиною, що буде свідчити про відсутність узгодження на етапах проектування. Відповідно, такий підхід призводить до зниження якості продукту та накопичення дефектів.

Список літератури

1. Ashmore, S., & Runyan, K. (2014). *Introduction to Agile Methods*. Addison-Wesley Professional.
2. Moran, A. (2015). *Managing Agile: Strategy, Implementation, Organisation and People*. Springer.
3. *Manifesto for Agile Software Development*. Retrieved May 13, 2020, from Manifesto for Agile Software Development: <https://agilemanifesto.org/>
4. Кош, М. (2011). *Succeeding with Agile: Software Development Using Scrum*. Вільямс.

РОЗРОБКА ТЕЛЕГРАМ БОТА ДЛЯ УПРАВЛІННЯ ПАСАЖИРСЬКИМИ ПЕРЕВЕЗЕННЯМИ

Комков М.А.

Харківський національний автомобільно-дорожній університет, Харків

Вступ

В сучасному світі автоматизація процесів відіграє важливу роль у підвищенні ефективності та зручності різних видів діяльності. Управління пасажирськими перевезеннями є однією з таких сфер, де застосування сучасних технологій може суттєво покращити якість обслуговування та оптимізувати роботу транспортних компаній. Одним із інструментів, що здатен досягти цих цілей, є телеграм бот, розроблений для автоматизації процесів планування та обробки заявок на пасажирські перевезення.

Телеграм бот – це програмний додаток, що працює в месенджері Telegram і дозволяє автоматизувати різноманітні завдання через взаємодію з користувачами. Використання телеграм ботів набуває все більшої популярності завдяки їхній доступності, зручності та функціональним можливостям. Вони здатні здійснювати різні операції, від простих інформаційних запитів до складних інтеграційних процесів.

Ця стаття присвячена розробці телеграм бота для управління пасажирськими перевезеннями. У ній розглянуто основні етапи створення бота, включаючи аналіз предметної галузі, проектування системи, розробку і тестування програмного забезпечення, а також його впровадження у практичну діяльність. Особливу увагу приділено опису середовища розробки, проектуванню системи класів та реалізації основних функцій бота, таких як додавання, перегляд, видалення та фільтрація заявок на перевезення.

Мета даного дослідження – продемонструвати можливості телеграм бота у сфері пасажирських перевезень, показати переваги автоматизації процесів та визначити перспективи подальшого розвитку цієї технології. Розроблений бот є зручним інструментом для організації та управління пасажирськими перевезеннями, що дозволяє значно спростити процеси обробки заявок, підвищити ефективність роботи та покращити якість обслуговування пасажирів.

Аналіз предметної галузі

Аналіз предметної галузі є важливим етапом у розробці будь-якої інформаційної системи, зокрема телеграм бота для управління пасажирськими перевезеннями. Метою цього аналізу є глибоке вивчення специфіки галузі, виявлення основних вимог до системи та визначення ключових аспектів для

подальшої розробки. У нашому випадку предметною галуззю є організація пасажирських перевезень, яка охоплює планування маршрутів, обробку заявок від пасажирів, управління транспортними засобами та забезпечення високого рівня обслуговування.

Загальна характеристика пасажирських перевезень

Пасажирські перевезення – це комплекс заходів, спрямованих на забезпечення перевезення людей від одного пункту до іншого з використанням різних видів транспорту. Організація пасажирських перевезень включає кілька важливих аспектів:

Планування маршрутів – визначення оптимальних шляхів руху транспортних засобів з урахуванням пасажиропотоків, дорожніх умов та інших факторів. Це включає аналіз попиту на перевезення, розробку графіків руху та маршрутизацію.

Обробка заявок – прийом і обробка заявок від пасажирів, включаючи бронювання місць, продаж квитків та інформування про розклад руху. Це потребує ефективної системи управління заявками, яка дозволяє швидко реагувати на запити пасажирів.

Управління транспортними засобами – координація роботи транспортних засобів, забезпечення їх технічного обслуговування та контроль за дотриманням графіку руху. Це включає моніторинг стану транспорту, управління водіями та планування технічного обслуговування.

Обслуговування пасажирів – забезпечення комфорту та безпеки пасажирів під час поїздки, надання додаткових послуг (інформаційних, розважальних тощо). Це потребує забезпечення високого рівня сервісу та вирішення проблем пасажирів у разі виникнення.

Вимоги до системи

Для ефективного управління пасажирськими перевезеннями телеграм бот повинен відповідати ряду вимог, що забезпечать його функціональність, надійність та зручність у використанні. Основними вимогами до системи є:

Інтуїтивно зрозумілий інтерфейс – бот повинен мати зрозумілий та зручний інтерфейс, що дозволяє користувачам легко взаємодіяти з системою та швидко отримувати необхідну інформацію.

Можливість обробки заявок – бот повинен забезпечувати прийом, обробку та зберігання заявок на перевезення, а також можливість перегляду та видалення заявок за необхідності.

Перевірка даних – система повинна перевіряти коректність введених даних, таких як номери маршрутів, дати поїздки та інша інформація, що забезпечить точність та надійність даних у системі.

Фільтрація та пошук – бот повинен мати можливість фільтрувати заявки за різними критеріями, такими як номер маршруту та дата поїздки, що дозволить швидко знаходити необхідну інформацію.

Логування та моніторинг – для забезпечення надійності та безпеки системи необхідно реалізувати механізми логування та моніторингу, що дозволять відстежувати роботу бота, виявляти та виправляти можливі помилки.

Проектування та розробка телеграм бота

Одним із ключових етапів розробки телеграм бота є вибір середовища розробки та налаштування робочого оточення. В нашому випадку ми використовуємо мову програмування Python та бібліотеку `python-telegram-bot`, яка забезпечує всі необхідні функції для взаємодії з API Telegram.

Проектування системи класів включало визначення основних об'єктів і їх взаємодії. *Основними класами* стали:

`Bot`: клас, що відповідає за загальну логіку роботи бота.

`Booking`: клас для роботи із заявками пасажирів.

`Route`: клас, який містить інформацію про маршрути.

Основні функції бота включають:

Додавання заявки. Користувач може додати нову заявку, вказавши пункт призначення, номер маршруту, прізвище та ініціали пасажирів, а також бажану дату поїздки. Бот перевіряє коректність введених даних і додає заявку до системи.

Перегляд заявок. Користувач може переглянути всі наявні заявки. Бот виводить список заявок, включаючи інформацію про маршрут, пункт призначення, пасажирів та дату поїздки.

Видалення заявки. Користувач може видалити окрему заявку, вказавши її номер у списку. Бот видаляє відповідну заявку з системи.

Фільтрація заявок. Користувач може фільтрувати заявки за номером маршруту та датою поїздки. Бот виводить тільки ті заявки, які відповідають заданим критеріям.

Практична реалізація телеграм бота

Розробка телеграм бота для управління пасажирськими перевезеннями є комплексним процесом, що включає кілька ключових етапів: проектування, розробка, тестування та впровадження. У цій частині розглянемо основні аспекти практичної реалізації бота, включаючи опис інтерфейсу користувача, приклади використання та вирішення виявлених проблем.

Інтерфейс користувача є важливим елементом телеграм бота, оскільки саме через нього відбувається взаємодія з користувачами. Основною метою є створення інтуїтивно зрозумілого та зручного інтерфейсу, який дозволить користувачам легко виконувати необхідні дії, такі як додавання, перегляд, видалення та фільтрація заявок.

Головне меню. Після запуску бота користувачам пропонується головне меню, яке містить кілька основних опцій:

- додати заявку
- переглянути всі заявки

- видалити заявку
- видалити всі заявки
- фільтрувати заявки за маршрутом і датою

Приклади використання бота

Для ілюстрації можливостей телеграм бота розглянемо кілька прикладів його використання.

Додавання заявки. Користувач запускає бота та обирає опцію "Додати заявку". Він вводить наступні дані:

"Київ, 1, Іванов І.І., 2023-06-15".

Бот перевіряє дані та додає заявку до системи.

Перегляд заявок. Користувач обирає опцію "Переглянути всі заявки". Бот виводить список заявок:

1. Маршрут: 1, Пункт призначення: Київ, Пасажир: Іванов І.І., Дата: 2023-06-15

Видалення заявки. Користувач обирає опцію "Видалити заявку" та вводить номер заявки "1". Бот видаляє заявку зі списку.

Фільтрація заявок. Користувач обирає опцію "Заявки за маршрутом і датою" та вводить дані "1, 2023-06-15". Бот виводить тільки ті заявки, які відповідають цим критеріям.

Висновки

Розробка телеграм бота для управління пасажирськими перевезеннями показала, що використання сучасних технологій дозволяє значно спростити і автоматизувати багато процесів, пов'язаних з організацією перевезень. Впровадження такого бота може значно покращити якість обслуговування пасажирів, знизити навантаження на працівників і підвищити ефективність роботи транспортних компаній.

В подальшому розвиток системи може включати додавання нових функцій, таких як інтеграція з іншими сервісами для планування маршрутів, аналіз даних для оптимізації перевезень та підвищення рівня безпеки. Таким чином, телеграм бот стає важливим інструментом в управлінні пасажирськими перевезеннями і має великий потенціал для подальшого вдосконалення.

Література

1. Official Documentation of Python. [Електронний ресурс]. Режим доступу: <https://docs.python.org/3/>. Дата доступу: 15.04.2024
2. Official Documentation of python-telegram-bot [Електронний ресурс]. Режим доступу: <https://python-telegram-bot.readthedocs.io/>. Дата доступу: 15.04.2024
3. Real Python [Електронний ресурс]. Режим доступу: <https://realpython.com/>. Дата доступу: 15.04.2024

РОЗРОБЛЕННЯ КОНЦЕПЦІЇ СИСТЕМИ ЕЛЕКТРОНОГО НАВЧАННЯ

Олена ШАПОШНІКОВА¹, Ерік АЛЕКСЕЄВ²

¹Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaer@gmail.com

²Харківський національний автомобільно-дорожній університет, Харків.

У світі стрімкого технологічного розвитку та постійних змін в освіті, системи електронного навчання стають необхідним інструментом для навчання та саморозвитку. Завдяки їхній актуальності та доступності, вони стають важливою частиною освітнього процесу, забезпечуючи можливість отримання знань в будь-який час та в будь-якому місці. Системи електронного навчання наразі є дуже актуальними завдяки:

- гнучкості та доступності, тому що люди можуть отримувати знання з будь-якого місця, використовуючи лише інтернет та пристрій для доступу до навчального контенту. Це особливо важливо в умовах глобальної мобільності та віддалених форматів роботи.

- актуальності інформації, бо системи електронного навчання здатні оперативнo оновлювати свій контент, щоб відповідати актуальним тенденціям та новинам в науці, технологіях та інших сферах. Це забезпечує користувачам доступ до найновіших знань та навичок.

- інтерактивність та залучення – використання відео уроків, виконання тестів, інтерактивних завдань та форумів дозволяє користувачам не лише засвоювати теоретичні знання, але й застосовувати їх на практиці, що сприяє кращому засвоєнню матеріалу.

- ефективність та індивідуалізація – кожен учень може сконцентруватися на певних аспектах навчання, враховуючи свої сильні та слабкі сторони, вивчати матеріал власним темпом, що сприяє ефективному засвоєнню інформації та індивідуальному підходу до кожного учня.

Враховуючи ці обставини пропонується система електронного навчання «Udl» для задоволення вимог сучасного освітнього середовища, яка створює ефективну платформу для навчання. Її застосування розглядається в контексті освітніх установ, фахівців різних напрямлень, студентів та викладачів, створюючи простір для взаємодії та навчання в онлайн-середовищі.

Цільова аудиторія «Udl» включає [1]:

- вчителів та викладачів, які шукають зручний інструмент для створення та управління онлайн-курсами;

- фахівців, які прагнуть розвивати свої навички та вдосконалювати свої знання у сфері інформаційних технологій;

- студентів, які вже мають певний досвід у цій сфері та прагнуть вдосконалити свої знання та навички або початківці які хочуть вивчити технологію з нуля;

- професіоналів, які готові надавати підтримку та консультації студентам під час їхнього навчання.
- навчальні заклади, які шукають інтегроване рішення для управління курсами та спрощення навчального процесу.

Враховуючи актуальність задачі та переваги систем електронного навчання, було розроблено положення про концепцію системи автоматизації взаємодії викладачів та учнів відповідно до [2, 3], зміст якого полягає у наступному.

Для вчителів які прагнуть передати свої знання та учнів які мають на меті покращити свої навички або здобути нові знання та продовжувати розвиватися в своїй галузі, які працюють над вивченням матеріалів курсів та виконанням домашніх завдань, складанні тестів для контролю засвоєння вивченого матеріалу, які надає ментор система «Udl» є системою автоматизації процесу взаємодії викладачів та учнів. Система надає доступ до високоякісного електронного навчання в різних галузях, дозволяє учням дивитися відео-лекції, виконувати домашні завдання, складати тести, отримувати додаткову інформацію до пройдених занять та відслідковувати статуси, як окремих занять, так і прогрес за курсом (рис. 1).

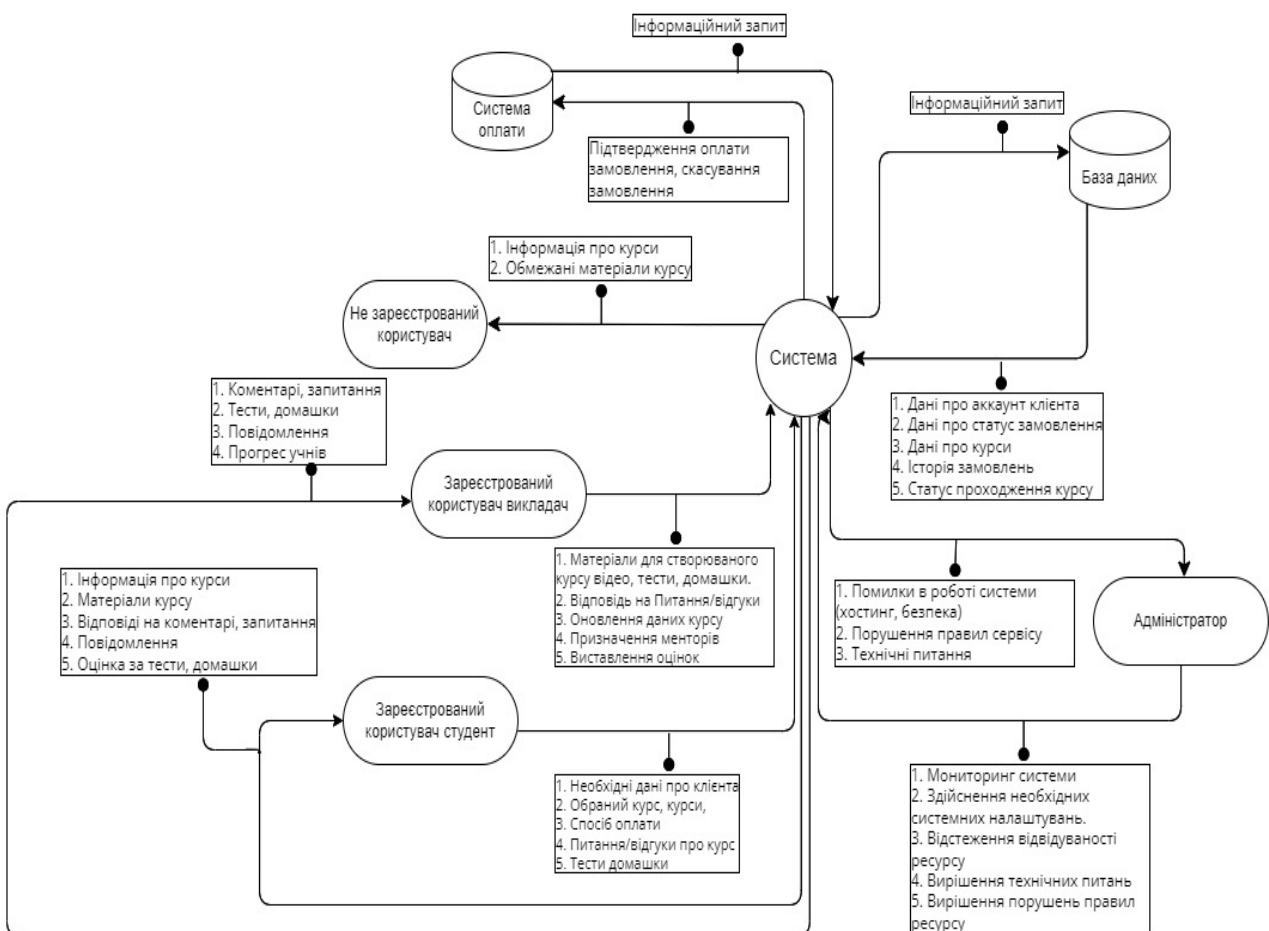


Рисунок 1 – Контексна Діаграма системи електронного навчання

На відміну від існуючих сайтів таких як Udemy, Stepik, ItGid система електронного навчання «Udl» надає наступні функції [4]:

- куратори можуть призначати менторів, які можуть відслідковувати прогрес учнів, давати рекомендації, передивлятися та оцінювати домашні завдання та тести.

- рекомендування курсів, категорію яких користувач частіше за все передивляється.

- відсутність фейкових дисконтних заходів.

Система електронного навчання «Udl» передбачає наявність повного процесу навчання для широкого набору курсів. Вона пропонує гнучкий графік навчання, що відповідає потребам користувача. Учні після успішного завершення курсів отримують сертифікати та дипломи.

Для швидкого, короткого і ємного опису призначення системи і її меж була розроблена контекстна діаграма (див. рис. 1) [5].

Основна мета контекстної діаграми - показати систему як одне ціле та її зв'язки з іншими системами, а також окремими користувачами або зовнішніми сутностями.

Як слідує з рисунка контекстна діаграма показує, як система взаємодіє зі своїм зовнішнім середовищем і встановлює контекст для подальшої розробки та аналізу.

Список використаних джерел

1. Стаття як обирати свою цільову аудиторію [Електронний ресурс] - Режим доступу: <https://www.seotm.com/ua/blog/kak-analizirovat-auditoriyu-saytaa.html>

2. Geoffrey A. Moore Marketing and Selling Disruptive Products to Mainstream Customers. Publisher: Harper Business; 3rd edition. 2014. 288 pages

3. Б. О. Котенко, О. П. Шапошнікова, О. В. Мнушка Розроблення прототипу мобільного навчаючого застосунку / Вісник ХНАДУ: зб. наук. пр. – Вип. 99. – Х., 2022. С. 170 – 175 С. 170-175

4. Аналіз сайтів конкурентів [Електронний ресурс] Режим доступу: <https://webtune.com.ua/statti/internet-marketing/analiz-sajtiv-konkurentiv/>

5. Застосунок Visual Paradigm [Електронний ресурс] – Режим доступу: <https://online.visual-paradigm.com/>

**ПЕРЕВАГИ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНИХ
СИСТЕМ В УПРАВЛІННЯ АВТОПАРКОМ**

Олена ШАПОШНІКОВА¹, Олександр ЗЕМЛЯНОЙ²

*¹Харківський національний автомобільно-дорожній університет, Харків,
Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaep@gmail.com*

²Харківський національний автомобільно-дорожній університет, Харків.

У сучасному світі ефективно управління автопарком із застосуванням інформаційних систем для управління основними видами діяльності стає все більш ключовим аспектом успіху для будь-якої організації, яка використовує транспортні засоби. Традиційні методи, що ґрунтувалися на використанні паперових документів та ручній обробці даних, нерідко виявлялися неефективними, витратними та схильними до помилок [1]. Однак завдяки впровадженню інформаційних систем управління автопарком ця галузь пройшла суттєву еволюцію, відкривши нові можливості для підвищення ефективності, зменшення витрат та загального зростання продуктивності. Тепер організації можуть використовувати різноманітні програмні засоби для автоматизації моніторингу, розподілу ресурсів, планування маршрутів, ведення обліку палива, управління технічним обслуговуванням та виконання інших важливих функцій. Це дозволяє підприємствам більш ефективно розподіляти ресурси та забезпечувати високу якість обслуговування [2]. Крім того, використання цифрових технологій дозволяє збільшити рівень безпеки та контролю над автопарком, що є важливим аспектом в умовах сучасного середовища. Таким чином, впровадження інформаційних систем в управління автопарком вирішує багато проблем, з якими стикаються організації, та сприяє їхньому стабільному розвитку та успіху.

Застосування цифрових технологій дозволяє здійснювати відстеження та моніторинг транспортних засобів, що забезпечує можливість відстежувати технічний стан транспортних засобів, та допомагає організаціям відповідати вимогам безпеки та нормативних актів. Дозволяє здійснювати ефективний контроль та приймати обґрунтовані рішення стосовно управління автопарком на основі актуальних даних. Тобто, дозволяє виявляти потенційні проблеми на ранніх стадіях та планувати необхідні роботи з обслуговування та ремонту, що знижує ризик непередбачених простоїв та подовжує термін експлуатації автопарку.

Наявність в інформаційних системах, централізованої бази даних та аналітичних інструментів дозволяє ефективно планувати та розподіляти ресурси.

В результаті автоматизації процесів збору та обробки даних у комп'ютеризованих системах суттєво скорочується паперова робота та підвищується точність даних. Це не лише зменшує обсяг рутинної роботи, а й мінімізує ризик людських помилок, забезпечуючи більш точну та своєчасну інформацію для прийняття рішень. Також автоматизовані системи мають інструменти для генерування детальних звітів. Зменшуються ризики людських помилок, тому що автоматизація процесів мінімізує ймовірність виникнення помилок, пов'язаних з ручним введенням та обробкою даних. Це дозволяє ефективно контролювати бюджет та виявляти можливості для економії ресурсів.

Разом з тим слід зауважити, що високі початкові витрати на впровадження комп'ютеризованих систем управління автопарком, що часто стає основною перешкодою, оскільки такі системи вимагають значних інвестицій у обладнання, програмне забезпечення та навчання персоналу. Складність інтеграції з іншими системами може виникнути в організаціях, які використовують кілька різних систем, а це потребуватиме їхньої інтеграції, яка може бути складною та, знову ж таки, потребувати додаткових витрат. При розробці таких систем слід також враховувати, що залежність від технологій та можливість технічних збоїв є потенційним ризиком, оскільки комп'ютеризовані системи можуть бути вразливими до збоїв, кібератак або втрати даних, що може призвести до перебоїв у роботі організації.

Не останніми мають бути задачі, пов'язані з конфіденційністю даних та необхідністю забезпечення безпеки, тобто враховувати все, що потребує захисту конфіденційної інформації [3].

Результати проведеного аналізу представлені у табл. 1.

Таблиця 1 – Переваги та недоліки цифровізації систем управління автопарком.

Переваги	Недоліки
Покращене відстеження та моніторинг транспортних засобів	Високі початкові витрати на впровадження
Ефективне планування та розподіл ресурсів	Залежність від технологій та можливість технічних збоїв
Покращене обслуговування та ремонт транспортних засобів	Проблеми з конфіденційністю даних та необхідність забезпечення безпеки
Скорочення паперової роботи та підвищення точності даних	Опір змінам з боку персоналу та необхідність додаткового навчання
Покращений контроль витрат та детальна звітність	Складність інтеграції з іншими системами
Підвищена безпека та відповідність нормативним вимогам	
Зменшення ризику людських помилок	
Своєчасне виявлення та усунення проблем	
Підвищення загальної ефективності логістичних операцій	

Враховуючи особливості розробки та впровадження комп'ютеризованих систем управління, розроблення інформаційної системи управління автопарком підприємства ґрунтується на концепції максимальної ефективності та простоти використання, яка спрямована на вирішення потреб невеликих автопарків. Основною метою вирішення поставленої задачі є створення інтуїтивно зрозумілого інструменту, який забезпечить ефективний облік та контроль за автомобілями, що перебувають в автопарку підприємства.

Основні задачі, які має виконувати інформаційна система управління автопарком полягають у наступному:

- простота та зручність інтерфейсу для забезпечення швидкої роботи операторів без потреби спеціального навчання;
- автоматизація процесів для ефективного обліку;
- гнучкість та адаптивність для врахування особливостей невеликих автопарків;
- надійність та безпека для захисту конфіденційних даних;
- аналітичні можливості для генерування звітів та статистики.

Таким чином впровадження інформаційних систем в управління автопарком в сучасних організаціях дозволяє забезпечити підвищення продуктивності, зниження витрат та покращення якості обслуговування. Хоча це принесе значні переваги, такі як покращений моніторинг та контроль транспортних засобів, ефективне планування ресурсів та скорочення паперової роботи, важливо також врахувати можливі недоліки, які включають високі початкові витрати та опір змінам з боку персоналу. Однак з правильним плануванням та уважним врахуванням цих аспектів, комп'ютеризація систем управління автопарком є важливим кроком для підвищення ефективності та успішного функціонування організації.

Список використаних джерел

1. Автоматизація бізнес-процесів: все про автоматизацію бізнесу. [Електронний ресурс] - Режим доступу: <https://bx-master.com/ua/news/biznes-protsessy/avtomatizatsiya-biznes-protsessov-vsye-ob-avtomatizatsii-biznesa>
2. Business process automation. [Електронний ресурс] – Режим доступу: <https://snov.io/glossary/business-process-automation/>
3. Недоліки автоматизації: що варто знати? [Електронний ресурс] - Режим доступу: <https://polaridad.es/uk/desventajas-de-la-automatizacion-que-debes-saber/>

ТЕЛЕГРАМ ЯК ІНСТРУМЕНТ ДЛЯ ДЕТАЛЬНОГО АНАЛІЗУ КОРИСТУВАЦЬКОЇ ВЗАЄМОДІЇ

Мацій Ольга Борисівна, доцент кафедри штучного інтелекту та програмного забезпечення

Харківського національного університету імені В.Н. Каразіна

Клочко Андрій Володимирович, студент факультета комп'ютерних наук Харківського національного університету імені В.Н. Каразіна

В сучасному цифровому світі, де кожна взаємодія користувача з технологією може бути виміряна та аналізована, здатність до глибокого розуміння цих даних стає вирішальною для успіху бізнесів та індивідуальних розробників. Аналіз користувацької взаємодії виходить за рамки простого вивчення поведінки на веб-сайтах чи в мобільних додатках; він охоплює кожен аспект спілкування між користувачем та цифровим продуктом, включаючи месенджери, як от Телеграм.

Незалежно від розміру та сфери діяльності, знання про те, як користувачі взаємодіють з їхніми продуктами чи послугами, є критично важливим. Це розуміння дозволяє як оптимізувати продукти та послуги для забезпечення кращого користувацького досвіду так і підвищити залученість користувачів через персоналізований контент та інтерактивність. На основі зібраних даних, можна виявляти та усувати проблеми у функціоналі продуктів, які можуть спричинити втрату клієнтів та планувати маркетингові стратегії.

При аналізі користувацької взаємодії через Телеграм, можна збирати різноманітні типи даних, кожен з яких надає унікальну інформацію про поведінку та переваги користувачів. Ось декілька прикладів таких даних:

- **Повідомлення.** Текстові повідомлення, які користувачі надсилають у чатах або через ботів, є основним джерелом даних. Аналіз цих повідомлень може допомогти зрозуміти, які запитання або проблеми найчастіше виникають у користувачів, які теми їх найбільше цікавлять, а також загальний тон та настрої спілкування.

- **Інтерактивні кнопки.** Інтерактивні кнопки в Телеграмі, такі як кнопки голосування, СТА (виклик до дії), або навігаційні кнопки в ботах, є потужним інструментом для збору даних про те, які дії користувачі виконують найчастіше. Аналіз натискань на ці кнопки може допомогти ідентифікувати найбільш ефективні виклики до дії та оптимізувати інтерфейс для кращого залучення користувачів.

- **Час та дата активності.** Реєстрація часу та дати, коли користувачі найактивніші, дозволяє зрозуміти піки взаємодії, що може бути важливо для планування розсилок, оновлень продукту або маркетингових кампаній.

Збір та аналіз даних в Телеграмі можуть стикатися з низкою викликів та проблем, що вимагають уважного підходу з боку компаній та розробників. Серед найбільш значущих питань є приватність даних та обробка великих об'ємів інформації.

Приватність даних є ключовим питанням в цифровому світі, особливо коли мова йде про платформи для спілкування, такі як Телеграм. Платформа зобов'язується захищати конфіденційність своїх користувачів, що може обмежувати доступ до деяких типів даних для аналізу. Компаніям і розробникам важливо дотримуватися правил приватності та регуляцій, таких як Загальний регламент про захист даних (GDPR) в Європі, щоб уникнути порушень прав користувачів на конфіденційність.

Telegram самостійно надає користувачам розширену статистику для публічних каналів, відкриваючи доступ до цінної інформації, яка може допомогти зрозуміти поведінку аудиторії та ефективність контенту. Власники каналів можуть отримати детальні дані про зростання підписників, загальну кількість сповіщень, а також перегляди окремих публікацій протягом години, що дає уявлення про найбільш активні періоди доби. Також доступна інформація про джерела переглядів та підписок, що дозволяє аналізувати, як користувачі знаходять канал. [1]

Попри широкий спектр можливостей для аналізу користувацької взаємодії, які Телеграм надає своїм користувачам, сама платформа не зосереджена на активному наданні глибоких аналітичних інструментів. Це спонукало до появи зовнішніх інструментів та сервісів, які дозволяють компаніям та індивідуальним розробникам збирати, аналізувати та інтерпретувати дані про взаємодію користувачів у Телеграмі. Ці інструменти відкривають нові можливості для моніторингу та оптимізації комунікації та маркетингових стратегій.

Одним з таких інструментів є Brand24. Він надає такі показники, як кількість онлайн-згадок, аналіз настроїв, охоплення та оцінка репутації. Існують і подібні до нього реалізації, які також дають змогу відслідковувати іншу корисну активність. Ці інструменти розширюють можливості аналізу в Телеграмі, надаючи користувачам багатий набір даних для покращення стратегій комунікації та маркетингу [2].

Важливість розробки та впровадження сервісів для збору статистики в Телеграмі залишається незаперечною. Ці інструменти відіграють ключову роль у допомозі компаніям та розробникам розуміти потреби та переваги своєї аудиторії, оптимізувати продукти та комунікації, а також підвищити загальну ефективність своєї присутності в Телеграмі. Однак, як і будь-який процес збору та аналізу даних, це супроводжується рядом викликів, які необхідно враховувати.

Джерела посилань

1. Вбудована статистика для каналів Telegram: 10 графіків. <https://onlypult.com/blog/built-in-statistics-for-telegram>
2. Аналітика каналу Telegram: найкращі безкоштовні та платні інструменти на 2024 рік. <https://collaborator.pro/blog/telegram-analytics-tools>

ПЕРЕВАГИ ВПРОВАДЖЕННЯ «ЕСО FRIENDLY» У АВТОМАТИЗАЦІЮ СИСТЕМИ ЗАМОВЛЕННЯ

Олена ШАПОШНИКОВА¹, Олег НАСАТОВ²

¹Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaep@gmail.com

¹Харківський національний автомобільно-дорожній університет, Харків.

У сучасному світі, де екологічні питання стають все більш актуальними, виникає потреба в інноваційних рішеннях, спрямованих на зниження екологічного впливу підприємств і організацій. Одним з таких рішень є система автоматизації замовлень, з екологічною дружністю.

Одним з основних аспектів екологічної спрямованості цієї системи є відмова від традиційного друку чеків. Замість цього, інформація про замовлення буде зберігатися в базі даних, що дозволить уникнути використання паперу, а також зменшити споживання енергії, пов'язане з роботою принтерів [1].

Основна ідея цієї системи полягає в збереженні всієї необхідної інформації про замовлення в електронному вигляді. При оформленні замовлення, дані про нього будуть внесені в базу даних, включаючи деталі про товари, кількість, ціни та відомості. Весь процес замовлення та оплати буде здійснюватися електронним шляхом, що спростить процес і зменшить час, витрачений на оформлення замовлення.

Одним з важливих аспектів системи є зручність і швидкість пошуку інформації про замовлення. Завдяки зберіганню даних в базі даних, оператори зможуть швидко знаходити необхідну інформацію про замовлення, дату та час замовлення. Це спростить роботу персоналу і покращить обслуговування клієнтів і матиме менший вплив на оточення.

Крім екологічних переваг, система автоматизації замовлень також може мати позитивний економічний вплив на підприємство. Зменшення витрат на папір, принтери та інші матеріали, пов'язані з традиційним процесом друку чеків, дозволить знизити витрати компанії.

Крім того, ефективність процесу оформлення замовлень підвищиться завдяки автоматизованій системі, що може призвести до збільшення обсягу продажів та покращення задоволеності клієнтів. Був проведений порівняльний аналіз характеристик традиційних систем та систем з еко-компонентами, результат якого представлений в табл. 1.

Недоліки автоматизованої системи:

– Витрати на впровадження: Впровадження еко-автоматизованої системи може потребувати значних первинних капіталовкладень.

– Технічна складність: Еко-автоматизована система може бути більш складною у налаштуванні та обслуговуванні.

Таблиця 1.2 – Порівняльна характеристика систем

Критерій	Традиційна система	Автоматизована система з еко-компонентами
Екологічність	Використовує папір та енергію для друку чеків	Не використовує папір, економить енергію
Зберігання інформації	Паперові чеки	Електронна база даних
Швидкість пошуку інформації	Повільний, потребує ручного пошуку	Швидкий, пошук за допомогою програмного забезпечення
Економія коштів	Витрати на папір, принтери, картриджі	Зниження витрат на папір, принтери, картриджі
Ефективність	Залежить від людського фактора	Автоматизований процес, менша ймовірність помилок
Задоволеність клієнтів	Можуть бути проблеми через затримки	Швидкий та зручний процес

Екологічна автоматизована система має багато переваг перед традиційною системою, таких як екологічність, економія коштів, економія часу, краща ефективність та задоволеність клієнтів.

Еко-френдлі системи автоматизації замовлень, які відмовляються від паперових чеків, стають все більш популярними у різних сферах, адже вони не лише економлять кошти, але й роблять значний внесок у захист довкілля. Прикладами їхнього успішного впровадження є наступні:

- McDonald's: у 2018 році McDonald's UK оголосив про те, що повністю відмовиться від паперових чеків у своїх 1361 ресторанах. Це рішення дозволило їм економити 100 тон паперу щороку;

- Starbucks: Starbucks також активно впроваджує еко-френдлі систему, де клієнти можуть отримати електронний чек на свій телефон або взагалі відмовитися від нього;

- KFC: KFC в Україні також пропонує клієнтам можливість отримати електронний чек замість паперового;

- Готелі: деякі готелі також почали використовувати еко-френдлі системи. Клієнти можуть отримати електронний чек на свій email або через мобільний застосунок;

- Онлайн-магазини: онлайн-магазини не друкують чеки, що робить їх еко-френдлі за замовчуванням;

Навіть без еко-компонентів, система автоматизації замовлень може значно зменшити потребу в папері. Отже, система автоматизації замовлень з екологічною спрямованістю є інноваційним рішенням, яке сприяє зменшенню екологічного впливу підприємств і організацій. Відмова від традиційного друку чеків та збереження всієї необхідної інформації в базі даних дозволяє зменшити споживання паперу та енергії, а також покращити ефективність процесу оформлення замовлень. Крім екологічних переваг, система може мати позитивний економічний вплив на підприємство, знижуючи витрати та підвищуючи задоволеність клієнтів.

Список використаних джерел

1. Зменшення використання енергії [Електронний ресурс] Режим доступу: <https://fastercapital.com/content/Going-Green-with-Electronic-Checks--Embracing-Paperless-Transactions.html> .

2. Крім екологічних переваг [Електронний ресурс] Режим доступу: <https://eqibank.com/blog/digital-banking-and-the-environment-how-its-helping-to-reduce-waste/> .

3. The Eco Forums [Електронний ресурс] Режим доступу: <https://www.global-ecoforum.org/> .

СИСТЕМНИЙ АНАЛІЗ СИСТЕМИ КЛІЄНТ-БАНК

Онищенко Яніна

Харківський національний автомобільно-дорожній університет

Успішний економічний розвиток держави неможливий без добре розвиненої банківської системи, на стан якої впливає як внутрішня економічна і політична ситуація, так і зміни світової банківської системи. Ці зміни є наслідком глибинних макроекономічних процесів у світовій економіці (інтеграція, лібералізація, різкий науково-технічний підйом) і вони зачіпають всі банки, незалежно від рівня їх розвитку. Однією з основних особливостей сучасної вітчизняної банківської системи є стрімкий розвиток комп'ютерних і телекомунікаційних технологій, впровадження мережевих пристроїв, що скорочує час обробки інформації, дозволяє провести комплексну автоматизацію діяльності, розробити механізми дистанційного обслуговування клієнтів і запропонувати новий асортимент послуг. Крім того, раціоналізація платіжної системи в країні потребує суттєвого скорочення операцій із готівкою на користь безготівкових розрахунків.

Розвиток дистанційного банківського обслуговування має на меті скорочення в Україні готівкових розрахунків, залучення у банківську систему значного обсягу коштів населення. В останні роки український ринок дистанційного банківського обслуговування відбувається не так стрімко як в розвинутих країнах світу. Декілька років тому основною послугою, пропонованою банками у сфері дистанційного обслуговування, була послуга «Клієнт-Банкінг». Поступово система «Клієнт-банкінг» стала замінюватися системою «Інтернет-банкінг».

В даний час багато банківських установ надають клієнтам ці обидві послуги, інші не використовуються зовсім, або з обмеженим функціоналом. Розвитку інформаційних та інтернет-технологій в банківській сфері сприяє значна конкуренція у даному секторі діяльності. Інтернет в цьому аспекті відіграє важливу роль, оскільки він дозволяє значно розширити ринок послуг, якими може користуватись клієнт, при цьому враховуються всі види дистанційного обслуговування або Інтернет-банкінгу.

Система клієнт-банк – є системою котра реалізує електронне обслуговування клієнтів банку, юридичних осіб безпосередньо з їхнього офісу за допомогою засобів електронної пошти. Наявність такої системи дає змогу клієнтам не виходячи з офісу, з дому відправити в банк платіжне доручення, отримати інформацію щодо проходження платежу, стан поточного рахунку, а також документів проведених за рахунком у будь-який момент часу. Система клієнт-банк відноситься до систем віддаленого банківського обслуговування, у деяких банках навіть передбачена онлайн допомога при важкості з роботою додатку, чи іншим проблемами з ним.

Система клієнт-банк надає можливість клієнту виконувати наступні функції:

- Вводити та передавати у банк платіжні документи, отримувати з банку виписки, про стан поточного рахунку та квитанції про отримання платіжних документів.

- Після завершення дня, надавати клієнту заключну виписку про рух коштів по поточному рахунку.

- Авторизація документів: накладання на них цифрових підписів головного бухгалтера та директора на електронні платіжні документи.

- Здійснювати захист електронних платіжних документів від підробки, несанкціонованої зміни або ознайомлення з ними.

- Інформаційне забезпечення: ведення баз даних класифікаторів та довідників.

- Генерація ключів цифрового підпису та призначення повноважень підпису посадовим особам на документах, формування електронної картки клієнта зі "зразками підписів".

Застосування сучасних засобів захисту інформації та криптографічних технологій вирішує проблеми, пов'язані з забезпеченням юридичної відповідальності за сформовані та передані електронні документи, з їх захистом від передбачуваних та не передбачуваних перекручень, від нав'язування хибної інформації тощо.

Література

1. Josefsson S., Liusvaara I. Edwards-Curve digital signature algorithm. Internet Research Task Force, 2017. 60 p. URL: <https://datatracker.ietf.org/doc/html/rfc8032> (date of access: 14.06.2023).

АНАЛІЗ ЕФЕКТИВНОСТІ ПРОЦЕДУРНОЇ ГЕНЕРАЦІЇ ЛАБІРИНТІВ ДЛЯ МУЛЬТИПЛЕЄРНИХ 3D ВЕБ-ІГОР НА БАЗІ THREE.JS

*Мацій Ольга Борисівна, доцент кафедри штучного інтелекту
та програмного забезпечення*

Харківського національного університету імені В.Н. Каразіна

Романов Микита Вячеславович, студент факультета комп'ютерних наук

Харківського національного університету імені В.Н. Каразіна

Сучасна ігрова індустрія представлена веб-іграми, які дозволяють гравцям швидко отримати доступ до ігрового контенту без потреби встановлення додаткового програмного забезпечення. Однією з передових технологій для створення 3D веб-ігор є Three.js, яка робить можливим розробку високоякісних ефектів та складних тривимірних сцен безпосередньо у веб-браузері. Водночас, процедурна генерація контенту, а саме лабіринтів, відкриває нові можливості для збільшення варіативності ігрових світів та розширення геймплею, не потребуючи значних додаткових зусиль при розробці.

Процедурна генерація контенту – це метод автоматичного створення даних за допомогою алгоритмів. У сфері веб-ігор це означає генерування унікальних ігрових елементів, як-от лабіринти, ландшапти або випадкові події, без прямої участі розробників. Процедурна генерація дозволяє створювати велику кількість контенту з обмежених ресурсів, забезпечуючи високу різноманітність ігрового досвіду [1].

До переваг процедурної генерації можна віднести її здатність створювати практично необмежену кількість унікального ігрового контенту, скорочуючи таким чином час і витрати на розробку, а також підвищуючи реграбельність, пропонуючи нові ігрові сценарії з кожним проходженням. Однак він також має свої недоліки, такі як непередбачуваність якості згенерованих структур, складність у збалансуванні ігрового процесу та технічні проблеми, пов'язані з оптимізацією продуктивності.

Three.js – це високорівнева бібліотека на основі WebGL, що дозволяє створювати складні 3D-сцени у браузері. Розробка 3D веб-ігор з її допомогою має низку переваг, серед яких використання стандартних веб-технологій (HTML, CSS, JavaScript) для створення інтерактивного 3D-контенту, легка інтеграція з іншими веб-сервісами та платформами, а також доступ до великої спільноти розробників. Однак оптимізація продуктивності для забезпечення безперебійної роботи 3D-додатків у різних веб-браузерах і на різному обладнанні є складним завданням для розробників [2].

Розробка методології для оцінки якості генерації лабіринтів була спрямована на аналіз складності, різноманітності та здатності лабіринтів сприяти захопливому ігровому процесу. У цьому процесі брали участь відгуки користувачів та кількісні показники, включаючи час, потрібний для проход-

ження лабіринту, та кількість ігрових сесій. Аналіз продуктивності, особливо з використанням інструментів для відстеження ресурсів браузера та комп'ютера, виявив важливі відомості про вплив різних методів генерації на швидкість відгуку та ефективність системи.

Порівняння процедурно генерованих лабіринтів із статичними, створеними вручну, підтвердило переваги у відтворюваності та різноманітності, хоча і виявило проблеми з оптимізацією продуктивності. На основі зібраних даних було доопрацьовано методи генерації, щоб виправити ці проблеми, звернувши особливу увагу на відгуки користувачів та аналіз продуктивності. Це призвело до розробки адаптивних алгоритмів, які можуть динамічно змінювати складність лабіринту, враховуючи кількість гравців та їхні уподобання, оптимізуючи при цьому навантаження на систему [3].

Особлива увага приділялася оптимізації продуктивності, в тому числі через використання ресурсів браузера. Розробники впровадили такі техніки, як лінійне завантаження та інстанціювання об'єктів, щоб мінімізувати споживання пам'яті та час завантаження, що сприяло швидкому відображенню складних 3D-лабіринтів навіть на обмеженому обладнанні.

Покращення візуальних ефектів та анімації за допомогою Three.js дозволило створити більш реалістичні освітлення, тіні, віддзеркалення та заломлення світла, забезпечуючи захоплюючий 3D-досвід, який використовує весь потенціал веб-технологій.

На завершальному етапі тестування з використанням автоматизованих інструментів, таких як Lighthouse в Google Chrome, відіграло ключову роль у виявленні та усуненні вузьких місць в коді, гарантуючи, що кінцевий продукт відповідає високим стандартам якості та продуктивності. Цей процес підкреслив важливість гнучкого підходу до розробки, який може адаптуватися до потреб користувачів і технологічних обмежень.

Література

1. Procedural Content Generation For Games: A Survey. URL: https://www.researchgate.net/publication/262327212_Procedural_Content_Generation_for_Games_A_Survey
2. Three.js Documentation. URL: <https://threejs.org/docs/>
3. Evaluating The Impact of Procedurally Generated Content on Game Immersion. URL: https://www.researchgate.net/publication/319173732_Evaluating_the_Impact_of_Procedurally_Generated_Content_on_Game_Immersion

ОГЛЯД СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ ЗМІШУВАЧ ПЕРІОДИЧНОЇ ДІЇ

Соболев Денис

Харківський національний автомобільно-дорожній університет

Цей тип системи масового обслуговування Змішувач періодичної дії настільки поширений, що його розгляду присвячена велика кількість літератури.

У роботі [1] розглядається теорія і практика аналізу складних систем та імітаційного моделювання. Особлива увага приділяється розвитку практичних навичок моделювання та дослідження систем. Підручник містить матеріали з аналітичного та імітаційного моделювання систем. Більша частина книги присвячена імітаційному моделюванню систем як найбільш практичному способу дослідження складних систем. Розглянуто алгоритми імітаційного моделювання систем, що ґрунтуються на представленні процесів функціонування систем засобами масового обслуговування та мереж Петрі. Алгоритми імітаційного моделювання мереж масового обслуговування та мереж Петрі на основі універсальних мов програмування є дуже гнучкими. Наведено перелік програм мовою Object Pascal, які ілюструють процедурні та об'єктно-орієнтовані підходи до побудови імітаційних алгоритмів. Автори також обговорюють методи дослідження імітаційних моделей, зокрема методи оптимізації. Основи імітаційного моделювання систем громадського транспорту з використанням мови GPSS та її реалізації GPSS World описано в [2]; на низці прикладів показано, як можна будувати імітаційні моделі за допомогою інструменту GPSS World. Значну увагу приділено прикладам порівняння результатів імітаційного та аналітичного моделювання систем у Крім того, функціональні можливості GPSS World використано для чисельного інтегрування систем диференціальних рівнянь Колмогорова та обчислення ймовірностей станів. Розглянуто модель, яка використовується для дослідження усередненого потоку подій та апроксимації декомпозиційного потоку до найближчого потоку. Найважливіші теоретичні та практичні питання системного аналізу.

На основі єдиного системного підходу в [3] викладено методології, методи і процедури аналізу та синтезу складних систем управління. Розглянуто системні питання структурного аналізу складних систем управління, методи їх математичного опису, проблеми прийняття рішень з використанням інтерпретаційних методів та експериментального оцінювання. Проілюстровано аналіз сценаріїв і цілей на основі графів, алгоритми координації з використанням парадигми нечітких множин, системні питання прийняття рішень в умовах невизначеності. Основну увагу приділено використанню методів системного аналізу для створення складних систем управління (СУ), об'єктом яких є технічні додатки або промислові підприємства в цілому. Основними

завданнями є: визначення загальної структури системи, організація необхідних взаємодій між підсистемами та елементами; врахування впливу зовнішнього середовища, оптимізація структури системи та розробка оптимальних алгоритмів функціонування.

У [4] розглядаються принципи, методи, прийоми та застосування системного підходу для прийняття ефективних управлінських рішень. Особливу увагу приділено методології, інструментарію та моделюванню в системному аналізі, деталям системного бачення інформаційних процесів та явищ, методам системного аналізу в науці та практиці.

Таким чином, незважаючи на те, що існує багато навчальних матеріалів по імітаційним моделям, дана тема є дуже актуальною і цікавою з точки зору навчального процесу.

Література

1. Стеценко, І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко ; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2010. – 399 с.

2. Жерновий Ю. В. Ж 59 Імітаційне моделювання систем масового обслуговування: Практикум. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 307 с.

3. Системний аналіз складних систем управління: Навч. посіб. / А.П. Ладанюк, Я.В. Смітюх, Л.О. Власенко та ін. — К.: НУХТ, 2013. — 274 с.

4. Системний аналіз інформаційних процесів: Навч. посіб. / В. М. Варенко, І. В. Братусь, В. С. Дорошенко, Ю. Б. Смольников, В.О. Юрченко.– К.: Університет «Україна», 2013. – 203с.

АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ АВТОМАТИЗАЦІЇ ПЛАНУВАННЯ КОШТОРИСУ

Олена ШАПОШНИКОВА¹, Валерія ХАЛИМОН²

¹*Харківський національний автомобільно-дорожній університет, Харків, Україна, ORCID 0000-0002-0405-8205, e-mail: shaposhnikovaep@gmail.com*

²*Харківський національний автомобільно-дорожній університет, Харків.*

У сучасному світі, де швидкість і точність є ключовими показниками успіху, автоматизація процесів стає вкрай необхідною. Автоматизовані системи бюджетного планування надають установам та організаціям засоби для ефективного управління фінансами, забезпечують точність і раціоналізацію процесу.

Мета цієї роботи є дослідження автоматизованих систем бюджетного планування та виявлення їхніх переваг та недоліків, щоб визначити важливість впровадження таких систем для підвищення ефективності управління фінансами.

За традиційного бюджетування процес є дорогим і схильним до людських помилок. Наприклад, введення даних вручну може призвести до неточностей і затримок.[1] Крім того, без необхідних інструментів складно зберігати й аналізувати великі обсяги даних. Попри це вже багато установ та організацій використовують автоматизовані системи, проте існують проблеми, пов'язані з їхньою інтеграцією, складністю налаштування та вартістю впровадження. Крім того, відсутність стандартизації може ускладнити обмін даними між різними системами та відділами всередині установи.

Перевагами автоматизованих систем є:

- значна економія часу і сил у процесі планування та аналізу кошторисів;
- зниження ймовірності помилок і неточностей;
- можливість швидкого оновлення та коригування даних;
- можливість зберігати й аналізувати історичні дані, що дає змогу ухвалювати більш обґрунтовані рішення.[2]

Недоліками можна вважати:

- високі витрати на впровадження та обслуговування системи;
- необхідність навчання персоналу правильному використанню програмних засобів; [3]
- ризик залежності від технічних проблем і вразливостей.

Автоматизовані системи бюджетного планування можуть містити:

- інструменти для автоматизованого збору та обробки фінансових даних;
- модулі для прогнозування та аналізу витрат;
- засоби візуалізації та звітності;
- модулі для розподілу бюджету та контролю витрат;
- користувацький інтерфейс для введення та аналізу даних база даних для зберігання інформації про бюджети, витрати та інші фінансові показники;
- алгоритми обробки даних для планування та аналізу;
- модулі звітності для створення звітів та аналізу даних.

Автоматизована система бюджетного планування призначена для використання економістами, бухгалтерами, фінансовими аналітиками, менеджерами проєктів, фінансовими директорами та іншими фахівцями, які відповідають за фінансове управління організацією.[4]

Нині існує безліч систем автоматизації бюджетного планування, включно з SAP, Oracle і Microsoft Dynamics. Кожна з цих систем має свої унікальні особливості та переваги, але всі вони спрямовані на полегшення процесу планування та управління бюджетом.

1. SAP ERP є однією з найбільш відомих та використовуваних систем управління підприємством. Вона має модуль для планування кошторису, який дозволяє підприємствам створювати, аналізувати та відстежувати бюджети, витрати та прибуток. SAP ERP забезпечує інтегрований підхід до управління фінансами, виробництвом, продажами та іншими аспектами бізнесу.

2. Oracle Hyperion [5] - це інтегрована платформа для корпоративного планування та аналізу. Вона надає широкі можливості для розробки, управління та аналізу кошторисів, включаючи прогнозування, моделювання та звітність. Oracle Hyperion дозволяє підприємствам планувати свої фінансові ресурси з точністю та ефективністю.

3. Microsoft Dynamics - це набір програмних продуктів для автоматизації бізнес-процесів, включаючи управління фінансами та планування кошторису [6]. Деякі з продуктів Microsoft Dynamics, такі як Microsoft Dynamics 365 Finance, надають інструменти для побудови бюджетів, прогнозування та аналізу витрат.

4. BudgetMaestro - це хмарна платформа для автоматизації планування кошторису. Вона дозволяє підприємствам легко створювати, аналізувати та оновлювати бюджети та прогнози, використовуючи інтуїтивний інтерфейс та розширені аналітичні можливості.

5. Adaptive Insights є середовищем для корпоративного планування, яке надає інструменти для планування кошторису, прогнозування, моделювання та звітності. Вона дозволяє підприємствам легко адаптуватися до змін у внутрішньому та зовнішньому середовищі шляхом швидкого оновлення та аналізу фінансових даних.

Після огляду вже існуючих автоматизованих систем бюджетного планування постає питання: якщо такі системи автоматизації є різного виду з різними можливостями та функціями, лише частина підприємств та організацій ними користуються?

Проаналізувавши роботу кількох організацій та установ, можливо відмітити, що найменше такі системи використовуються у Державних установах. Це явище обумовлено багатьма факторами основними з яких є:

1. Відсутність адаптації до конкретних потреб: багато з цих систем були розроблені для великих або приватних компаній і можуть не враховувати специфічні вимоги державного сектору або місцеві умови.

2. Недостатня підтримка і навчання персоналу: впровадження нових технологій часто вимагає навчання персоналу. Якщо постачальник системи не надає належної підтримки або співробітники не можуть швидко освоїти нові інструменти, вони можуть виявитися не в змозі використовувати систему.

3. Відсутність фінансування на впровадження та підтримку: впровадження автоматизованої системи вимагає великих витрат. Якщо державні установи не отримують достатнього фінансування на ці цілі, вони можуть не впровадити такі системи.

4. Питання безпеки та конфіденційності: забезпечення високого рівня безпеки та конфіденційності особливо важливе для державних органів. Якщо системи бюджетного планування не відповідають вимогам щодо захисту конфіденційної інформації, їх використання може бути обмежене.

Ці фактори впливають на можливості використання автоматизованих систем бюджетного планування не тільки державних установ, а й різного виду бізнесу, інших установ та організацій.

Системи автоматизації бюджетного планування відіграють важливу роль у підвищенні ефективності фінансового управління. Вони скорочують час і ресурси, необхідні для планування та аналізу, а також забезпечують точність і достовірність даних. Однак вибір і впровадження системи мають бути ретельно адаптовані до специфіки установи та потреб її користувачів для успішного користування системами бюджетного планування кошторису.

Список використаних джерел

1. Бенько М. М. Інформаційні системи і технології в бухгалтерському обліку: монографія. Київ: Київський національний торговельно-економічний університет, 2010. 336 с.
2. Останкова Л.А. Аналіз, моделювання та управління економічними ризиками: навч. посібн. Київ.: Центр навчальної літератури, 2011. – 256 с.
3. Електронний ресурс (Конспект лекцій з навчальної дисципліни комп'ютеризація облікових систем) Режим доступу : https://elib.lntu.edu.ua/sites/default/files/elib_upload/D1%84%D0%B5%D0%B4%D1%96%D0%BA%202/page18.html
4. Скриньковський Р. Інформаційні технології в організації бухгалтерського обліку на підприємстві. Економіка. 2019. С. 10.
5. Електронний ресурс (Oracle Hyperion, Режим доступу: <https://www.oracle.com/ua/>)
6. Електронний ресурс (Microsoft Dynamics) Режим доступу: <https://www.microsoft.com/en-us/dynamics-365>
7. Електронний ресурс (Фінансова політика як складова економічного розвитку) Режим доступу: <https://knute.edu.ua/file/MzEyMQ==/3043ab09788d719c6ad51708e587adb2.pdf>

УДК 625.85

КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ СТРУКТУРИ АСФАЛЬТОБЕТОНУ ДЛЯ ДОСЛІДЖЕННЯ ЙОГО ВЛАСТИВОСТЕЙ ПІД ДІЮ НИЗЬКИХ ТЕМПЕРАТУР

Хомутенко Денис Геннадійович, аспірант

Харківський національний автомобільно-дорожній університет

Анотація. У статті описана методика комп'ютерного моделювання структури зразку асфальтобетону, після випробувань на низькотемпературну тріщиностійкість за методом ACCD. За результатами зроблені висновки щодо можливості використання даної методики для комп'ютерного моделювання структури зразку асфальтобетону та дослідження напружено-деформованого стану, що виникає в асфальтобетоні під час охолодження, а також складний напружено-деформований стан від дії експлуатаційного навантаження та низьких температур..

Ключові слова: асфальтобетон, тривимірне моделювання, комп'ютерне моделювання, структура асфальтобетону, низькотемпературні напруження

Вступ

До основних причин руйнувань асфальтобетонного покриття автомобільних доріг відноситься поперечне розтріскування, причиною якого зокрема є низькотемпературні напруження. Це в свою чергу призводить до зменшення довговічності асфальтобетонного покриття та комфорту користування автомобільними дорогами. Усунення таких дефектів вимагає витрати значних матеріальних ресурсів. Війна та складна економічна ситуація в Україні, робить проблему ресурсозбереження все більш актуальною. В таких умовах важливим є підвищення низькотемпературної тріщиностійкості асфальтобетону.

Низькотемпературні властивості асфальтобетону значною мірою залежать від його складу та структури. Основними складовими асфальтобетону є бітумне в'язуче (бітум, або бітум з модифікатором), мінеральний порошок, дрібний заповнювач (пісок, висівки), крупний заповнювач (щебінь), а також пори. Відповідно структура асфальтобетону поділяється на мікроструктуру, мезоструктуру та макроструктуру. Складові асфальтобетону мають різні властивості, зокрема температурний коефіцієнт об'ємного розширення та модуль пружності, які в свою чергу змінюються залежно від температури. Таким чином, асфальтобетон розглядається як багатокомпонентний, багатозфазний, грубодисперсний матеріал із неоднорідною структурою [1]. Тому його властивості пов'язані з характеристикою його складових та взаємодією між ними. Одним із шляхів покращення низькотемпературних властивостей асфальтобетону є оптимізації його складу та структури.

Враховуючи неоднорідність структури асфальтобетону та комплексний вплив факторів (експлуатаційні навантаження, фактори навколишнього середовища), що діють на асфальтобетонне покриття автомобільних доріг при їх експлуатації, доцільним є використання методів комп'ютерного моделювання для аналізу напружено-деформованого стану асфальтобетону при його охолодженні [2].

В свою чергу результати комп'ютерного моделювання потребують верифікації на основі експериментальних даних. Тому, для прогнозування низькотемпературних властивостей асфальтобетону та оптимізації його складу, актуальним є використання новітніх методів випробування асфальтобетону на низькотемпературну тріщиностійкість, наприклад методу ACCD [3], у поєднанні з методом комп'ютерного моделювання [4].

Аналіз публікацій

При комп'ютерному моделюванні структури асфальтобетону застосовуються двовимірні [1] або тривимірні моделі [2]. В свою чергу тривимірні моделі можуть бути простими геометричними моделями, моделями заснованими на методах випадкової генерації структури та моделями заснованими на методах тривимірної реконструкції [5].

Технологія тривимірної реконструкції структури асфальтобетону за допомогою методів цифрової обробки зображень порівняно з методами

випадкової генерації структури може краще відображати реальну морфологію заповнювача та структуру асфальтобетону [5].

Метод тривимірної реконструкції заснований на аналізі двомірних зображень перерізів асфальтобетону. Для аналізу структури асфальтобетону можуть бути використані цифрові камери, скануючий електронний мікроскоп і рентгенівська комп'ютерна томографія [6].

До переваг методу комп'ютерної томографії можна віднести неруйнівний характер дослідження, наявність розвинутих програмних комплексів для тривимірної реконструкції. В свою чергу до недоліків можна віднести дороговизну обладнання для промислової комп'ютерної томографії та наявність іонізуючого випромінювання [6].

В свою чергу, до переваг методу скануючої електронної мікроскопії можна віднести отримання зображень високої роздільної здатності. В свою чергу до недоліків можна віднести дороговизну обладнання та руйнівний характер дослідження, так як необхідний контакт з поверхнею зразка [6].

Перевагами методу обробки цифрових зображень, отриманих за допомогою цифрових камер, є простота, доступність, точність і низька вартість обладнання. До недоліків можна віднести руйнівний характер дослідження, так як необхідна видимість поверхні зразка, складність у дослідженні мікроструктури та визначені пор [6].

Враховуючи складну економічну ситуацію в Україні, а також той факт, що зразок асфальтобетону вже пошкоджено при проведенні дослідження на низькотемпературну тріщиностійкість за методом ACCD, доцільно сконцентрувати увагу на методі обробки цифрових зображення, отриманих за допомогою цифрових камер.

Мета та постановка задачі

Метою даної роботи є розробка методики комп'ютерного моделювання структури асфальтобетону, зразків, що пройшли випробування на низькотемпературну тріщиностійкість за методом ACCD. Розробка методики направлена на підвищення точності визначення напружено-деформованого стану та механізму розвитку дефектів в асфальтобетоні при його охолодженні. Отримання реалістичної моделі надасть можливість подальшої оптимізації складу і структури асфальтобетону, та підвищення його низькотемпературної тріщиностійкості.

Методи та об'єкти дослідження

Об'єктами дослідження являються зразки асфальтобетону, що пройшли випробування на низькотемпературну тріщиностійкість за методом ACCD [3].

Для проведення фотографування перерізів зразку асфальтобетону використовується цифровий фотоапарат CANON EOS 2000D, з об'єктивом CANON EF-S 18-55mm f/4-5.6 IS STM. Щоб отримати фотографії високої якості використовується стенд, що забезпечує постійну орієнтацію фотографій та складається з столику з закріпленим фотоапаратом, кільцевої світлодіодної

лампи з температурним діапазоном 4000–5700 К, інтервалометра, площадки для розміщення зразку асфальтобетону з фотограмметричними маркерами.

Отримання послідовної серії перерізів зразку асфальтобетону, шляхом поетапного зрізання його шарів, виконується на верстаті з числовим програмним керуванням для обробки каменю. Такі верстати забезпечують досить високу точність обробки близько 0,01 мм.

Результати дослідження

Пропонується розглянути методику комп'ютерного тривимірного моделювання структури асфальтобетону на основі реконструкції за двовимірними цифровими зображеннями послідовної серії перерізів зразку асфальтобетону, що пройшов випробування на низькотемпературну тріщиностійкість за методом АССД. Алгоритм складається з семи етапів, що наведені на рис. 1.



Рис. 1. Схема алгоритму комп'ютерного моделювання структури асфальтобетону

Етап 1. Випробування зразку асфальтобетону на низькотемпературну тріщиностійкість за методом ACCD. На даному етапі проводиться дослідження зразку асфальтобетону на низькотемпературну тріщиностійкість відповідно до методики [3]. Зразки асфальтобетону можуть бути виготовлені з лабораторно приготованої асфальтобетонної суміші, переформованої асфальтобетонної суміші або ж відібраними з асфальтобетонного покриття.

Етап 2. Отримання послідовної серії перерізів зразку асфальтобетону. Зразок асфальтобетону, після випробувань на низькотемпературну тріщиностійкість за методом ACCD, має тріщину у визначеному місці. В цьому разі проведення реконструкції його структури неруйнівними методами має меншу актуальність, так як основні дослідження вже виконані, а зразок пошкоджено. В такому випадку зразок поміщається в фрезерний верстат з числовим програмним керуванням для обробки каменю. За допомогою зазначеного верстата послідовно видаляються шари зразку асфальтобетону, сегментуючи його.

Етап 3. Фотографування серії перерізів зразку асфальтобетону. На даному етапі виконується фотографування кожного перерізу зразку асфальтобетону. Фотографування виконується на спеціальному стенді, що забезпечує високу якість отриманих фотографій та їх орієнтування у просторі.

Етап 4. Обробка цифрових зображень. Отримані фотографії повинні розташовуватися в системі координат XYZ для можливості проведення подальшої якісної тривимірної реконструкції. Для цього пропонується використати методи фотограмметрії, що реалізовані в програмному комплексі Reality Capture.

Далі фотографії обробляють шляхом перетворення зображення у градації сірого, а далі бінаризація [5; 6; 7; 8] з подальшим усуненням включень і незначних за розміром частинок, а також розмиванням та фільтрацією зображення [8]. Наступними є сегментація та спрощення геометрії за допомогою відповідного програмного забезпечення [5].

Етап 5. Тривимірна реконструкція структури асфальтобетону. В подальшому оброблені та сегментовані цифрові зображення серії перерізів зразку асфальтобетону експортують в програмні комплекси для тривимірної реконструкції зображень такі як DragonFly, 3D Slicer, XCAT, Avizo, та інші. В яких виконується тривимірна реконструкція структури асфальтобетона, а також проводиться спрощення геометрії моделі для подальшого експорту в систему автоматизації інженерних розрахунків.

Етап 6. Чисельне моделювання. На цьому етапі виконується експорт отриманої тривимірної моделі в систему автоматизації інженерних розрахунків, такі як ANSYS, NASTRAN, Abaqus FEA, COMSOL Multiphysics, FEATool Multiphysics, та інші. Та подальший розрахунок за допомогою методу скінченних елементів або методу дискретних елементів [5].

Етап 7. Інтерпретація та аналіз отриманих даних. Верифікацію отриманих результатів тривимірного моделювання структури асфальтобетону, для зразків, що були виготовлені з лабораторно приготованої асфальтобетонної

суміші, можливо провести порівнявши їх з даними гранулометричного складу визначених в лабораторних умовах. Результати розрахунків напружено-деформованого стану порівнюють з результатами лабораторних випробувань. Також, можливе порівняння результатів чисельного моделювання отриманих з використанням тривимірної моделі за даною методикою та іншими моделями (геометричної моделі Радовського, моделями виконаними за допомогою методів випадкової генерації структури: на основі системи простих кульок, на основі діаграм Вороного, та інших).

Доцільним також буде дослідження характеру тріщини від дії низьких температур, що утворилася в зразку асфальтобетону при випробуванні АССД. При цьому можливо дослідити залежність розвитку тріщини від розподілу пор в зразку асфальтобетону, а також наявності інших дефектів структури зразку асфальтобетону.

Висновки

Запропонована методика комп'ютерного тривимірного моделювання структури зразку асфальтобетону є перспективним напрямком дослідження. Використання даної методики надасть можливість більш точно дослідити напружено-деформований стан, що виникає в асфальтобетоні під час охолодження, а також складний напружено-деформований стан від дії експлуатаційного навантаження та низьких температур.

Подальші наукові дослідження доцільно направити на практичну апробацію вказаної методики-та верифікацію отриманих результатів.

Література

1. Маляр В. В. Розрахунок напружено-деформованого стану асфальтобетону на основі моделювання його структури. Вісник ХНАДУ. 2014. № 67. С. 98–101. URL: <https://dSPACE.khadi.kharkov.ua/items/cba2168a-f333-4656-8f80-22e26799ce0e> (дата звернення: 11.04.2024).
2. Маляр В. В. Визначення температурних напружень в асфальтобетоні методом комп'ютерного моделювання. Вісник ХНАДУ. 2021. № 93. С. 105–111. URL: <https://doi.org/10.30977/BUL.2219-5548.2021.93.0.105> (дата звернення: 11.04.2024).
3. Evaluation of Low Temperature Cracking Resistance of WMA / S.-S. Kim та ін. Athens OH : The Ohio Department of Transportation Office of Statewide Planning & Research, 2015. 121 с. URL: <https://www.researchgate.net/publication/279942111> (дата звернення: 21.03.2024).
4. Хомутенко Д. Г. Новітні методи експериментального визначення низькотемпературних властивостей асфальтобетону: виклики та перспективи для України. Збірник матеріалів XIV Міжнародної науково-технічної конференції аспірантів та молодих учених "НАУКОВА ВЕСНА 2024", м. Дніпро, 27 берез. 2024 р. Дніпро, 2024. URL: <https://rmv.nmu.org.ua/ua/arkhiv-zbirok-konferentsiy/naukova-vesna.php> (дата звернення: 15.04.2024).
5. Improved Procedure for the 3D Reconstruction of Asphalt Concrete Mesostructures Considering the Similarity of Aggregate Phase Geometry between Adjacent CT Slices / C. Wang та ін. Materials. 2023. № 16 (1). С. 234. URL: <https://doi.org/10.3390/ma16010234> (дата звернення: 11.04.2024).
6. Alawneh M., Soliman H. Using Imaging Techniques to Analyze the Microstructure of Asphalt Concrete Mixtures: Literature Review. Applied Sciences. 2023. № 13 (13). С. 7813. URL: <https://doi.org/10.3390/app13137813> (дата звернення: 11.04.2024).

7. Bessa I. S., Castelo Branco V. T., Soares J. B. Evaluation of different digital image processing software for aggregates and hot mix asphalt characterizations. *Construction and Building Materials*. 2012. № 37. С. 370–378. URL: <https://doi.org/10.1016/j.conbuildmat.2012.07.051> (дата звернення: 11.04.2024).

8. Klimczak M., Jaworska I., Tekieli M. 2D Digital Reconstruction of Asphalt Concrete Microstructure for Numerical Modeling Purposes. *Materials*. 2022. № 15(16). С. 5553. URL: <https://doi.org/10.3390/ma15165553> (дата звернення: 12.04.2024).

ПІДХІД ДО СИСТЕМНОГО АНАЛІЗУ ІМІТАЦІЙНОЇ МОДЕЛІ ГЛОБАЛЬНОГО ПОТЕПЛІННЯ

Чмутов Артур

Харківський національний автомобільно-дорожній університет

Метою роботи є системний аналіз математичних моделей, який дозволить побудувати моделі, що відображають сучасний стан глобального потепління та прогнозують майбутнє потепління на основі показників, дозволяючи спостерігати за змінами ситуації під впливом різних факторів.

Математичні моделі глобального потепління є складними системами. Складна система - це відкрита система, яка постійно змінюється внутрішньо, під впливом зовнішнього середовища і в результаті дії як внутрішніх, так і зовнішніх факторів. Моделювання є найбільш ефективним способом дослідження складних систем, як на етапі проектування, так і в процесі експлуатації, для різноманітних цілей, включаючи технічні, економічні, екологічні, соціальні та інформаційні. Це найпотужніший і універсальний метод дослідження та оцінки ефективності різних систем, на поведінку яких впливає вплив випадкових факторів.

Дана робота спрямована на дослідження складних систем, які є імітаційними моделями. Модель імітує основні явища, з яких складається функціонування системи, і описує об'єкт дослідження певною мовою, зберігаючи при цьому його логічну структуру, часовий порядок, особливості та конфігурацію інформації про стан процесу. Опис компонентів реальної системи в імітаційній моделі має певний логіко-математичний характер і являє собою набір алгоритмів, які імітують функціонування цієї системи.

Просторові інвентаризації парникових газів мають на меті визначити місцезнаходження джерел викидів та оцінити обсяги їхніх викидів. На відміну від традиційних інвентаризацій на рівні країни, просторові інвентаризації враховують характеристики джерел викидів і коефіцієнти викидів, характерні для кожного джерела, що дозволяє будувати більш детальні інвентаризації. Геоінформаційні системи, створені для просторової інвентаризації парникових газів у видобувних галузях, є моделюючими комплексами, які реалізують ці процедури інвентаризації газів.

Ратифікувавши Кіотський протокол, країни взяли на себе зобов'язання щорічно проводити інвентаризацію викидів парникових газів на національному рівні та подавати свої оцінки до Рамкової конвенції ООН про зміну клімату в національному звіті про інвентаризацію. Було розроблено єдиний стандарт звітності, і країни домовилися про те, які джерела викидів оцінювати.

Розвинені країни подають свої статистичні звіти у відкритому доступі, де їх може прочитати будь-хто. У просторових інвентаризаціях важливо мати не лише географічні координати джерел викидів, але й статистичні дані про видобувні та виробничі потужності кожного джерела. На рівні окремих джерел викидів інформація недоступна (цей тип інформації є статистичною таємницею). Натомість великі компанії публікують різноманітні річні звіти, в яких оприлюднюють інформацію про свою діяльність та результати роботи. Інформація про їхню діяльність також доступна на офіційному веб-сайті компанії. Статистичний облік в Україні не є добре організованим, значна частина інформації є статистичною таємницею, а деякі дані, необхідні для інвентаризації викидів парникових газів, не збираються. За допомогою математичних моделей викиди парникових газів від окремого джерела можна оцінити лише за наявності достатньої статистичної інформації про це джерело викидів.

Література

1. Ладанюк А.П. Системний аналіз складних систем управління - К.: НУХТ, 2006. - 227 с.
2. Державна служба статистики України [Електронний ресурс]. URL: <http://ukrstat.gov.ua/> (дата доступу: 07.03.2024).

ЯК ШТУЧНИЙ ІНТЕЛЕКТ ЗМІНИТЬ АВТОТРАНСПОРТНІ ВАНТАЖОПЕРЕВЕЗЕННЯ

Чорнобай Е.І., Карпішен Б.С.

Харківський національний автомобільно-дорожній університет, Харків

Технологія штучного інтелекту (ШІ) існує з 1950-х років, але сьогодні набуває все більшої популярності завдяки технологічному прогресу. Вона дозволяє комп'ютерним програмам навчатися та імітувати людську поведінку і має здатність імітувати такі характеристики, як здатність міркувати, розуміти сенс і вчитися на власному досвіді. Як результат він може розумно виконувати завдання без вказання прямих інструкцій.

Сьогодні вважається, що ШІ знижує кількість людських помилок і його можна побачити всюди навколо нас. Якщо користуватись такими системами, як Siri або Alexa, то були кроки до використання цієї технології. На даний момент

використовується в багатьох сферах та галузях, включаючи рекламу, охорону здоров'я, роздрібну торгівлю та електронну комерцію, фінансування, виробництво, вантажоперевезення та транспорт [1].

У США кілька великих компаній розгорнули автономні вантажівки для перевезення вантажів, що принесло країні численні переваги, включно з надзвичайно важливою – зменшення впливу вантажного транспорту на навколишнє середовище та оптимізацію доставки та переміщення вантажів, що зменшує затори на дорогах і допомагає при грошових втратах.

Штучний інтелект змінив не лише роботу бек-офісу, а й життя на дорозі. При згадці цього терміну ШІ люди зазвичай думають про автономні вантажівки, також відомі як автономні саморухливі вантажівки. Хоча такі компанії, як Tesla і Uber, лідирують у розробці автомобілів, що самі рухаються, ШІ присутній на дорогах вже сьогодні.

Всі можуть погодитися з тим, що життя на дорозі стало простіше і безпечніше завдяки системам ADAS. Сьогодні ці системи передової допомоги водію, які призначені полегшити керування автомобілем і підвищити безпеку на дорозі. Ця система використовує сучасні технології, такі як датчики, камери, лідари та радіочастотні модулі, для надання водієві інформації та втручання в керування автомобілем у важких або небезпечних ситуаціях[2]. ADAS оснащує транспортні засоби поєднанням сенсорних технологій і алгоритмів обробки штучного інтелекту, щоб відчувати навколишнє середовище навколо автомобіля, обробляти його, а потім або надавати інформацію водієві, або вживати заходів.

ШІ вже почав змінювати світову економіку і, ймовірно, продовжить це робити. За оцінками аналітиків, розвиток штучного інтелекту може збільшити світову економіку приблизно на 13 трильйонів доларів США до 2030 року. Це включає транспортний сектор, де ШІ, як очікується, спричинить додаткові зриви. Згідно з оцінками світових дослідницьких компаній, у 2017 році світовий ринок технологій штучного інтелекту, пов'язаних із транспортом, досяг 1,2–1,4 мільярда доларів.

Він може зрости до 3,1–3,5 мільярда доларів США до 2023 року, зареєструвавши зведений річний темп зростання (CAGR) від 12 до 14,5 відсотків протягом 2017–2023 років. Швидке зростання цього ринку зумовлене багатьма перевагами. Штучний інтелект може забезпечити транспорт, зокрема підвищення ефективності, безпеки водія та зниження витрат. У 2017 році найбільша частка припадає на Північну Америку на світовому ринку перевезень ШІ – 44 відсотки [3].

Технології штучного інтелекту роблять логістичну галузь більш автоматизованою та орієнтованою на користувача.

Коли штучний інтелект поєднується із аналізом великих даних, Інтернетом речей та машинним навчанням, можна розробити ефективні рішення. Ось деякі переваги використання ШІ у транспорті:

- Відстеження транспортних засобів

- Ефективне використання ємності сховища
- Скорочення експлуатаційних витрат
- Оптимізація маршруту для логістики
- Зниження дорожньо-транспортних пригод
- Підвищення стійкості
- Скорочення часу в дорозі

Обчислювальна складність алгоритмів III. Алгоритм — це набір правил, спрямованих на вирішення певної проблеми. Розробка та аналіз алгоритмів є важливою частиною технологій III. Він вимагає, щоб бути ефективним з точки зору меншого часу, необхідного для обробки алгоритму для великої кількості вхідних даних.

У транспорті це включає такі проблеми, як маршрутизація транспортних засобів та оптимальне планування для водіїв та учасників дорожнього руху. Складність обчислень обмежує методи III, оскільки більшість алгоритмів класифікуються як NP-проблема та NP-повна задача. У глибокому навчанні багато прихованих рівнів побудовано в архітектурі мережі. Отже, складність залежить від того, коли великі дані містять шум і спотворення, з яких важко витягнути функції. У транспорті дані можна збирати з багатьох джерел, починаючи від датчиків на дорозі, підключених пристроїв, платних пунктів, GPS до хмарних додатків тощо.

Збір та аналіз даних з різних джерел відкривають нові можливості для вдосконалення транспортної інфраструктури та управління транспортним потоком. Сучасні технології дозволяють використовувати великі обсяги даних (Big Data) та застосовувати аналітичні методи, включаючи машинне навчання та штучний інтелект, для вивчення взаємозв'язків та розробки ефективних стратегій управління рухом.

Важливим аспектом є розробка алгоритмів, які можуть працювати в реальному часі та приймати рішення на основі поточних даних про рух та інші важливі параметри. Це включає у себе не лише оптимальний вибір маршруту для конкретного транспортного засобу, але й прогнозування змін у руховому потоці та адаптацію стратегій управління в реальному часі.

Ці джерела зберігають великі дані про різні функції транспорту. Прикладами таких характеристик є транспортний потік, швидкість, наповненість і поведінка мандрівників. Отже, це ускладнює процес обчислення для вирішення конкретної проблеми [4]. Ось декілька головних джерел:

- Датчики на дорозі та інфраструктура;
- Системи глобального позиціонування (GPS) в транспортних засобах;
- Інформація, яка збирається на платних ділянках доріг;
- Хмарні додатки, що збирають дані про рух та маршрути в реальному часі;
- Системи відеоспостереження на дорогах;
- Додатки для смартфонів та веб-платформи, які збирають дані про рух, подорожі;

- Дані від автовиробників.

Наведемо декілька прикладів як нейронні мережі змінюють автотранспортні вантажоперевезення і транспортний потік в цілому.

Розроблена глибока нейронна мережа для прогнозування потоку трафіку до 60 хвилин. Дані про транспортні потоки були зібрані з автострад по всій Каліфорнії. Був використаний неконтрольований стек автокодерів під назвою SAE model, і навчено за допомогою жадібного покрокового алгоритму. Він виділяє важливі характеристики потоку трафіку, оскільки кожен вихід надходить у мережу як вхід. Потім для прогнозування стабільності маршруту застосовується контрольований рівень логістичної регресії.

Модель продемонструвала чудову продуктивність для нелінійної кореляції просторових і часових даних трафіку, порівняно з іншими методами машинного навчання, такими як опорна векторна машина (SVM) і багаторівнева нейрона мережа і нейронна мережа зворотного розповсюдження. У той час як було запропоновано використати неконтрольовану мережу глибоких перекохань (DBN), навчену за допомогою жадібного пошарового алгоритму для вивчення важливих функцій із шаблону потоку. Для прогнозування застосовано контрольований рівень регресії. Однак пропонувалось багатозадачний регресійний рівень із методом групування ваги, щоб об'єднати кілька завдань разом і навчити модель. У подальшому це показало підвищення точності моделі. Крім того, розробили модель глибокого навчання з використанням польових даних, зібраних у Каліфорнії, для фіксації довгострокового прогнозу потоку транспорту. Модель була успішною у вилученні просторових характеристик потоку трафіку за допомогою гурткової нейронної мережі та характеристик тимчасового потоку трафіку з повторюваної нейронної мережі [5].

Також можна привести приклад математичної моделі та алгоритму рішення багатоіндексної задачі з використанням ШІ.

Транспортна проблема - це проблема вирішення маршруту транспортування товару між місцем виробництва товару та місцем, де він потрібен, і безпечної доставки товару до місця попиту. У традиційних транспортних задачах усі попит і пропозиція, пов'язані з транспортуванням, і транспортні витрати є вже визначеними величинами.

Наприклад, обсяг продукції на місці виробництва продукту дорівнює U , місце виробництва – i , кількість продукції, необхідна місці попиту, – G , а місце попиту – j . Коли кількість місць попиту перевищує кількість місць виробництва, то планування транспортних витрат T і кількості M перевезеної продукції виглядає наступним чином:

$$T = M * g. \quad (1)$$

Тут g – ціна одиниці перевезення товару. Потім генетичний алгоритм використовується для розрахунку вантажу кількома маршрутами, а потім маршрут транспортування обирається шляхом порівняння вантажу. Якщо є

лише одне місце виробництва X , але є два місця попиту, існує два транспортні маршрути Y_1 та Y_2 , а кількість товарів, необхідна для місць попиту Y_1 та Y_2 , становить G_1 та G_2 відповідно. Тоді є дві транспортні схеми. Перший — відправити всі товари з двох місць попиту з місця виробництва, а потім транспортувати їх між двома місцями попиту. Другий – транспортувати товар до двох місць попиту. Кінцевою метою транспортної проблеми є зменшення вантажу; тоді шлях першої схеми перевезень розраховується наступним чином:

$$\begin{aligned} T_1 &= G_1 * g + G_2 * g, \\ T_2 &= M * g + (M - G_1) * g, \\ M &< U. \end{aligned} \tag{2}$$

Таким чином, якщо $T_1 < T_2$, буде обрано маршрут транспортування варіанту 1, а якщо $T_1 > T_2$, то для перевезення товару буде обрано маршрут транспортування варіанту 2. Крім того, з розвитком сучасного суспільства транспортні проблеми є складними, тому нам потрібно поєднати технологію штучного інтелекту та створити математичні моделі та алгоритми для більш ефективного вирішення багато індексних транспортних проблем. У той же час він також може забезпечити якість предметів у процесі транспортування, так що витрати на транспортування можуть бути мінімізовані. Звичайно, під час транспортування виникнуть круті схили та повороти, що збільшить ризик транспортування. Таким чином, планування транспортного маршруту також має враховувати транспортний ризик. Формула розрахунку транспортного ризику виглядає наступним чином:

$$\text{Risk} = \frac{n + t}{L} * 100\%. \tag{3}$$

Тут n означає кількість крутих схилів, які зустрічаються на транспортному маршруті, тоді як t представляє кількість кривих на транспортному маршруті, а L означає загальну довжину транспортного маршруту. Тому багато індексно транспортна задача повинна не тільки мінімізувати фрахт, але й зменшити фактор ризику, щоб забезпечити найменшу втрату вартості товару під час транспортування [6].

Висновки. Оскільки вантажоперевезення є важливою частиною логістики та транспортної індустрії, людям цікаво дізнатися, що майбутнє ШІ та машинного навчання означає для них. У міру того, як все більше вантажних компаній впроваджуватимуть нові методи роботи, ШІ продовжуватиме впливати як на водіїв, так і на власників самоскидів.

Він продовжить впроваджувати інновації у те, як далекобійники орієнтуються на дорозі, як вони керують робочим процесом та регулюють операції. Незважаючи на те, що ШІ розвивається, він не може зробити все. Людське

втручання, як і раніше, необхідне і, як і раніше, знаходиться на вершині ланцюга. У той час як багато представників автотранспортної галузі приймають ШІ, в інших він викликає побоювання.

Перелік посилань

1. Konieczna I. Artificial intelligence in transport – will chat GPT replace the forwarding department [Електронний ресурс] / Iwona Konieczna // Prilo. – 2023. – Режим доступу до ресурсу: <https://prilo.com/artificial-intelligence-in-transport-will-chat-gpt-replace-the-forwarding-department/>.
2. Карпішен Б. С. АНАЛІЗ РОЗРОБКИ І ВИКОРИСТАННЯ СИСТЕМИ ADAS В АВТОМОБІЛІ / Карпішен.Б.С. // Комп'ютерні технології і мехатроніка. Збірник наукових праць за матеріалами III міжнародної науково-методичної конференції. – Харків, ХНАДУ, 2022.. – 2022. – №3. – С. 29–33.
3. Conde, Maria Lopez, and Ian Twinn. "How Artificial Intelligence Is Making Transport Safer, Cleaner, More Reliable and Efficient in Emerging Markets." (2019): n. pag. Web.
4. Badalian V. AI has a finger on the pulse of the transport industry [Електронний ресурс] / Vartan Badalian // GreenBiz Group. – 2023. – Режим доступу до ресурсу: <https://www.greenbiz.com/article/ai-has-finger-pulse-transport-industry>.
5. W. Huang, G. Song, H. Hong and K. Xie, "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning," in IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 5, pp. 2191-2201, Oct. 2014, doi: 10.1109/ TITS.2014. 2311123.
6. Junfang Cao, "Mathematical Model and Algorithm of Multi-Index Transportation Problem in the Background of Artificial Intelligence", Journal of Advanced Transportation, vol. 2022, Article ID 3664105, 11 pages, 2022. <https://doi.org/10.1155/2022/3664105>

УДК 004.42:659

ПРОГНОЗУВАННЯ ЕФЕКТИВНОСТІ ПРОСУВАННЯ САЙТУ ЗА ДОПОМОГОЮ МОВИ ПРОГРАМУВАННЯ ТА АНАЛІЗУ ДАНИХ R

*Кривінченко Д. Р., студент, Мельников О. Ю., канд. техн. наук, доцент
Донбаська державна машинобудівна академія*

Анотація. Проаналізовано показники оцінки ефективності просування сайту, сформульовано модель прогнозування цієї ефективності, яка складається з чотирьох частин. Наведено розрахунки за цими моделями методами багатомірного регресійного аналізу та штучних нейронних мереж у середовищі R.

Ключові слова: просування сайту, моделювання, штучні нейронні мережі, мова R.

Основними елементами представництва комерційних компаній в мережі Інтернет є сайти. Це можуть бути сайти-візитки, сайти-брошури, сайти-профілі

компанії (з детальною інформацією про компанію), промо-сайти (спрямовані на рекламу певного товару, послуги або бренду), сайти-вітрини (де розміщуються новини компанії) тощо [1]. Однак у всіх випадках ці сайти потрібно «просувати», тобто проводити рекламну кампанію для залучення відвідувачів.

Методи просування сайту – це комплекс дій, які підвищують рейтинг сайту в пошукових системах і привабливість для цільової аудиторії. Просування сайту сприяє загальному розвитку ресурсу. Ця діяльність включає як внутрішні фактори, що знаходяться під контролем власника сайту (наприклад, якість тексту і дизайну сторінок, поліпшення структури та навігації тощо), так і зовнішні фактори (обмін посиланнями, розміщення в каталогах та інші заходи, спрямовані на збільшення та стимулювання кількості та частоти посилань на ресурс) [2–4].

Оцінка ефективності просування сайту містить аналіз показників, які свідчать про те, наскільки успішно проводиться просування й чи досягаються поставлені цілі. Деякими з основних показників, які варто враховувати при оцінці ефективності просування сайту, є наступні [5]:

- позиції в пошукових системах (чим вище позиція сайту по ключових словах у пошукових системах, тим більше ймовірність, що користувачі знайдуть його й перейдуть на сайт);

- трафік на сайті (показник загальної кількості відвідувачів на сайті: чим більше трафіку, тим більше потенційних клієнтів або користувачів, які можуть залишити заявку або зробити покупку);

- конверсії (показник кількості відвідувачів сайту, що зробили якусь цільову дію, наприклад, залишили заявку на сайті або зробили покупку: чим вище конверсія, тим більше ефективність просування);

- середній час на сайті (скільки часу користувач проводить на сайті: чим більше часу вони проводять, тим більше ймовірність, що сайт їм цікавий і вони зроблять якусь цільову дію);

- відмови (скільки користувачів залишають сайт без здійснення якоїсь цільової дії: чим менше відмов, тим більше ефективність просування);

- соціальна активність (скільки користувачів діляться контентом сайту в соціальних мережах: чим більше активності, тим більше людей довідаються про сайт та ймовірність, що вони перейдуть на нього).

У роботі [5] було розроблено модель прогнозування з наступними вхідними факторами:

- місце сайту в пошукових системах (Seng);
- кількість публікацій в соціальних мережах (Post);
- кількість банеропоказів або показів контекстної реклами (Show).

Вихідні фактори моделі:

- загальна кількість відвідувачів сайту (Traf);
- кількість відвідувачів сайту, які зробили якусь цільову дію (Conv);
- середній час, який відвідувач провів на сайті (Time).

Усі фактори наведено за добу. Приблизний вигляд даних представлено на рис. 1.

	A	B	C	D	E	F	G
1	Date	Seng	Post	Show	Traf	Conv	Time
2	01.фев	25	15	540	1103	65	110
3	02.фев	25	13	531	1026	80	106
4	03.фев	25	15	593	1500	84	108
5	04.фев	24	12	588	1063	71	106
6	05.фев	24	13	579	1189	84	101
7	06.фев	23	11	539	1250	59	102
8	07.фев	20	12	526	1103	53	101
9	08.фев	20	12	519	1411	80	104
10	09.фев	20	12	513	1169	61	102
11	10.фев	21	11	599	1358	64	104
12	11.фев	21	12	530	1485	66	100
13	12.фев	22	13	566	1050	65	109
14	13.фев	23	13	552	1213	86	103

Рисунок 1 – Вхідні дані

Можна прогнозувати як окремі фактори, так і сукупні результати. Для цього дані потрібно «нормалізувати» до діапазону від 0 до 1, а потім розрахувати їхню суму (рис. 2).

	A	B	C	D	E	F	G	H	I	J	K
1	Date	Seng	Post	Show	Traf	Conv	Time	TrafP	ConvP	TimeP	Sum
2	01.фев	25	15	540	1103	65	110	0,1677	0,2917	1	1,45938
3	02.фев	25	13	531	1026	80	106	0,0063	0,6042	0,6	1,21046
4	03.фев	25	15	593	1500	84	108	1	0,6875	0,8	2,4875
5	04.фев	24	12	588	1063	71	106	0,0839	0,4167	0,6	1,10052
6	05.фев	24	13	579	1189	84	101	0,348	0,6875	0,1	1,13551
7	06.фев	23	11	539	1250	59	102	0,4759	0,1667	0,2	0,84256
8	07.фев	20	12	526	1103	53	101	0,1677	0,0417	0,1	0,30938
9	08.фев	20	12	519	1411	80	104	0,8134	0,6042	0,4	1,81758
10	09.фев	20	12	513	1169	61	102	0,3061	0,2083	0,2	0,71441
11	10.фев	21	11	599	1358	64	104	0,7023	0,2708	0,4	1,37314
12	11.фев	21	12	530	1485	66	100	0,9686	0,3125	0	1,28105
13	12.фев	22	13	566	1050	65	109	0,0566	0,2917	0,9	1,24827
14	13.фев	23	13	552	1213	86	103	0,3983	0,7292	0,3	1,42749
15	14.фев	23	12	589	1209	99	104	0,3899	1	0,4	1,78994

Рисунок 2 – Дані після нормалізації

Математично постановку задачі можна записати так:

x_1 – Seng (позиція);

x_2 – Post (кількість);

x_3 – Show (кількість);

y_1 – Traf (кількість);

y_2 – Conv (кількість);

y_3 – Time (секунд).

y_s – сумарний показник.

У роботі [5] була поставлена та реалізована задача створення інформаційної системи – спеціального додатка для оцінки ефективності просування сайту методом штучних нейронних мереж. Але цікаво порівняти різні модифікації цієї моделі. Тому для проведення розрахунків було обрано мову програмування та аналізу даних R [6].

Ми маємо чотири моделі прогнозування:

$$y_1 = f(x_1, x_2, x_3) \quad (1)$$

$$y_2 = f(x_1, x_2, x_3) \quad (2)$$

$$y_3 = f(x_1, x_2, x_3) \quad (3)$$

$$y_s = f(x_1, x_2, x_3) \quad (4)$$

Для кожної моделі створено окремий скрипт, який виконує розрахунки побудовою багатofакторної лінійної регресії та штучної нейронної мережі, а також дозволяє визначити найкращу архітектуру нейронної мережі та більш ефективний метод її навчання для певного набору даних.

Результати розрахунків методом багатofакторної лінійної регресії за чотирма моделями зведено у табл. 1.

Таблиця 1 – Результати розрахунків методом багатofакторної регресії

N	Seng	Post	Show	Traf	res1	error1	Conv	res2	error2	Time	res3	error3	Sum	res4	error4
1	25	15	540	1103	1195	0.083643	65	65.40	0.006202	110	105.5	0.0409502	1.4594	1.211	0.17037
2	25	13	531	1026	1172	0.142114	80	68.53	0.143393	106	106.3	0.0031224	1.2105	1.310	0.08244
3	25	15	593	1500	1215	0.189946	84	73.35	0.126825	108	105.7	0.0208443	2.4875	1.443	0.41985
4	24	12	588	1063	1198	0.126960	71	79.11	0.114280	106	106.5	0.0046964	1.1005	1.602	0.45592
5	24	13	579	1189	1205	0.013149	84	75.53	0.100859	101	106.0	0.0496588	1.1355	1.493	0.31513
6	23	11	539	1250	1184	0.052428	59	73.81	0.251052	102	106.2	0.0407646	0.8426	1.430	0.69668
7	20	12	526	1103	1234	0.119008	53	69.04	0.302673	101	104.0	0.0299207	0.3094	1.221	2.94650
8	20	12	519	1411	1232	0.127110	80	67.99	0.150093	104	104.0	0.0001103	1.8176	1.190	0.34513
9	20	12	513	1169	1229	0.051671	61	67.09	0.099890	102	104.0	0.0192142	0.7144	1.164	0.62927
10	21	11	599	1358	1237	0.089358	64	82.41	0.287733	104	105.4	0.0130293	1.3731	1.638	0.19281
11	21	12	530	1485	1221	0.177853	66	69.84	0.058123	100	104.6	0.0458598	1.2811	1.266	0.01185
12	22	13	566	1050	1230	0.170972	65	73.19	0.125993	109	104.9	0.0379480	1.2483	1.382	0.10682
13	23	13	552	1213	1209	0.002959	86	71.29	0.171091	103	105.3	0.0227341	1.4275	1.348	0.05597
14	23	12	589	1209	1213	0.003479	99	79.07	0.201325	104	106.0	0.0188243	1.7899	1.579	0.11768
15	24	12	588	1215	1198	0.014026	94	79.11	0.158363	108	106.5	0.0139091	2.0983	1.602	0.23641
16	25	13	577	1046	1189	0.136723	72	75.42	0.047542	106	106.6	0.0051967	1.0857	1.512	0.39256
17	25	13	554	1430	1180	0.174538	51	71.98	0.411290	102	106.4	0.0435383	1.0532	1.411	0.33975
18	25	13	538	1023	1174	0.148022	53	69.58	0.312787	110	106.4	0.0330506	1.0417	1.341	0.28730
19	21	11	596	1288	1236	0.040737	52	81.97	0.576256	108	105.3	0.0246231	1.3764	1.625	0.18043
20	20	14	540	1140	1260	0.104900	55	66.67	0.212103	105	103.2	0.0170438	0.8286	1.143	0.37988
21	20	10	525	1137	1214	0.067553	97	73.37	0.243648	110	104.9	0.0464026	2.1973	1.356	0.38309
22	20	12	580	1098	1254	0.142497	92	77.14	0.161575	107	104.3	0.0254196	1.7114	1.458	0.14822
23	21	12	559	1190	1232	0.035071	99	74.18	0.250681	109	104.7	0.0392239	2.2501	1.393	0.38091
24	21	15	592	1271	1274	0.002519	79	72.42	0.083328	101	103.6	0.0253917	1.2032	1.329	0.10474
25	22	13	538	1301	1219	0.062991	55	68.99	0.254417	101	104.7	0.0369290	0.7661	1.259	0.64309
26	23	13	514	1252	1195	0.045369	51	65.59	0.286094	105	105.2	0.0015236	0.9801	1.181	0.20497
27	23	12	516	1358	1186	0.126727	90	68.13	0.243025	108	105.6	0.0221408	2.3148	1.259	0.45601
28	26	12	530	1054	1147	0.087779	74	70.81	0.043097	109	107.3	0.0155017	1.4442	1.403	0.02870
Середня помилка						0.09072			0.1937			0.02491			0.3826

Як можна побачити, модель №1 забезпечує середню помилку, яка дорівнює 0.09072, модель №2 – 0.1937, модель №3 – 0.02491, модель №4 – 0.3826. Це означає, що метод багатofакторної лінійної регресії дуже добре підходить для прогнозування середнього часу, якій відвідувач провів на сайті, та загальної кількості відвідувачів сайту, не дуже – для кількості відвідувачів сайту, які зробили якусь цільову дію, та зовсім не може бути застосований до сумарного показника.

Далі будемо застосовувати метод штучних нейронних мереж. Після численних запусків скрипта моделі №1 для різних параметрів кількості прихо-

ваних шарів та кількості нейронів у них з'ясовано, що найкращий результат (кореляція – 0,7254; середня абсолютна приведена помилка – 0,1873) забезпечує перцептрон з одним прихованим шаром та п'ятьма нейронами у ньому (рис. 3, рис. 4).

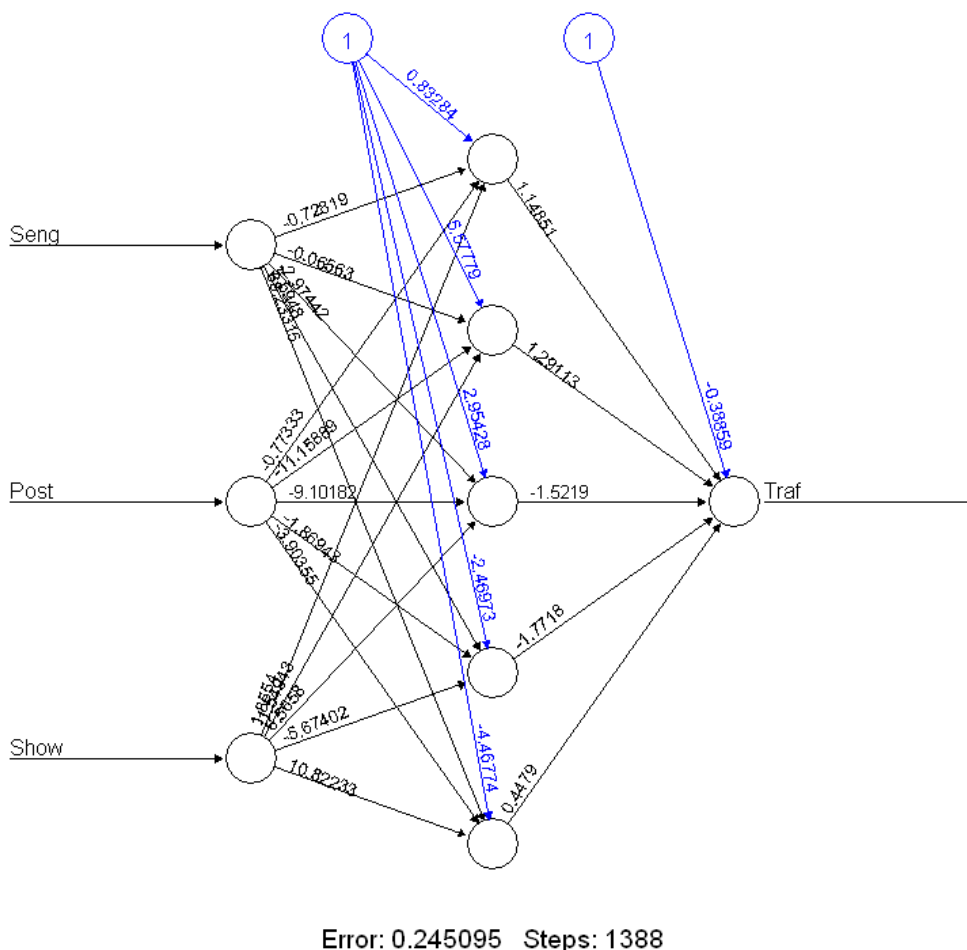


Рисунок 3 – Граф нейронної мережі моделі №1

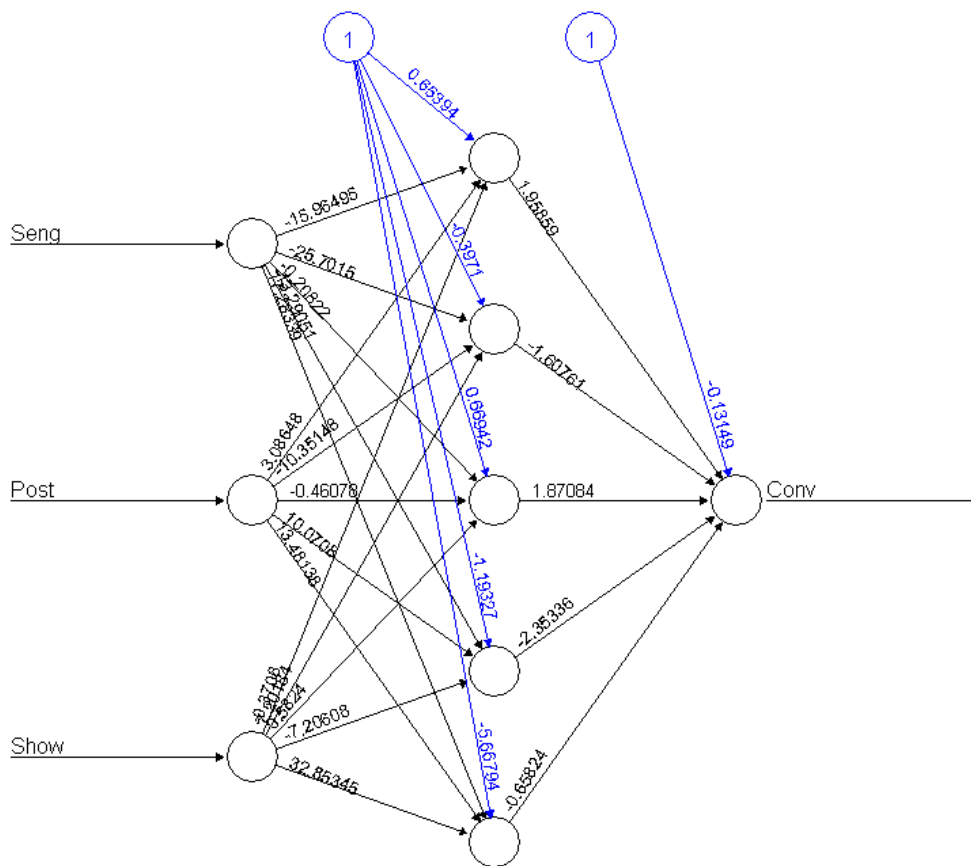
```
> w2
```

	Seng	Post	Show	Traf	Conv	Time	Sum	res	error
1	25	15	540	1103	65	110	1.4594	1171	0.14223
2	25	13	531	1026	80	106	1.2105	1041	0.03224
3	25	15	593	1500	84	108	2.4875	1411	0.18721
9	20	12	513	1169	61	102	0.7144	1370	0.42194
14	23	12	589	1209	99	104	1.7899	1074	0.28211
15	24	12	588	1215	94	108	2.0983	1073	0.29750
19	21	11	596	1288	52	108	1.3764	1357	0.14460
20	20	14	540	1140	55	105	0.8286	1161	0.04301
27	23	12	516	1358	90	108	2.3148	1422	0.13516

Рисунок 4 – Результати розрахунків за моделлю №1

Після запусків скрипта моделі №2 для різних параметрів кількості прихованих шарів та кількості нейронів у них з'ясовано, що найкращий результат

(кореляція – 0,7574; середня абсолютна приведена помилка – 0,305) також забезпечує перцептрон з одним прихованим шаром та п'ятьма нейронами у ньому (рис. 5, рис. 6).



Error: 0.362168 Steps: 602

Рисунок 5 – Граф нейронної мережі моделі №2

```

w2
> w2

```

	Seng	Post	Show	Traf	Conv	Time	Sum	res	error
1	25	15	540	1103	65	110	1.4594	61.58	0.0712
3	25	15	593	1500	84	108	2.4875	70.59	0.2795
5	24	13	579	1189	84	101	1.1355	73.13	0.2264
8	20	12	519	1411	80	104	1.8176	50.31	0.6185
10	21	11	599	1358	64	104	1.3731	47.52	0.3432
11	21	12	530	1485	66	100	1.2811	63.64	0.0491
14	23	12	589	1209	99	104	1.7899	77.00	0.4583
20	20	14	540	1140	55	105	0.8286	46.26	0.1820
27	23	12	516	1358	90	108	2.3148	65.18	0.5170

Рисунок 6 – Результати розрахунків за моделлю №2

Після запусків скрипта моделі №3 для різних параметрів кількості прихованих шарів та кількості нейронів у них з'ясовано, що найкращий результат (кореляція – 0,679; середня абсолютна приведена помилка – 0,3952) забезпечує

персептрон з двома прихованими шарами та трьома нейронами у кожному (рис. 7, рис. 8).

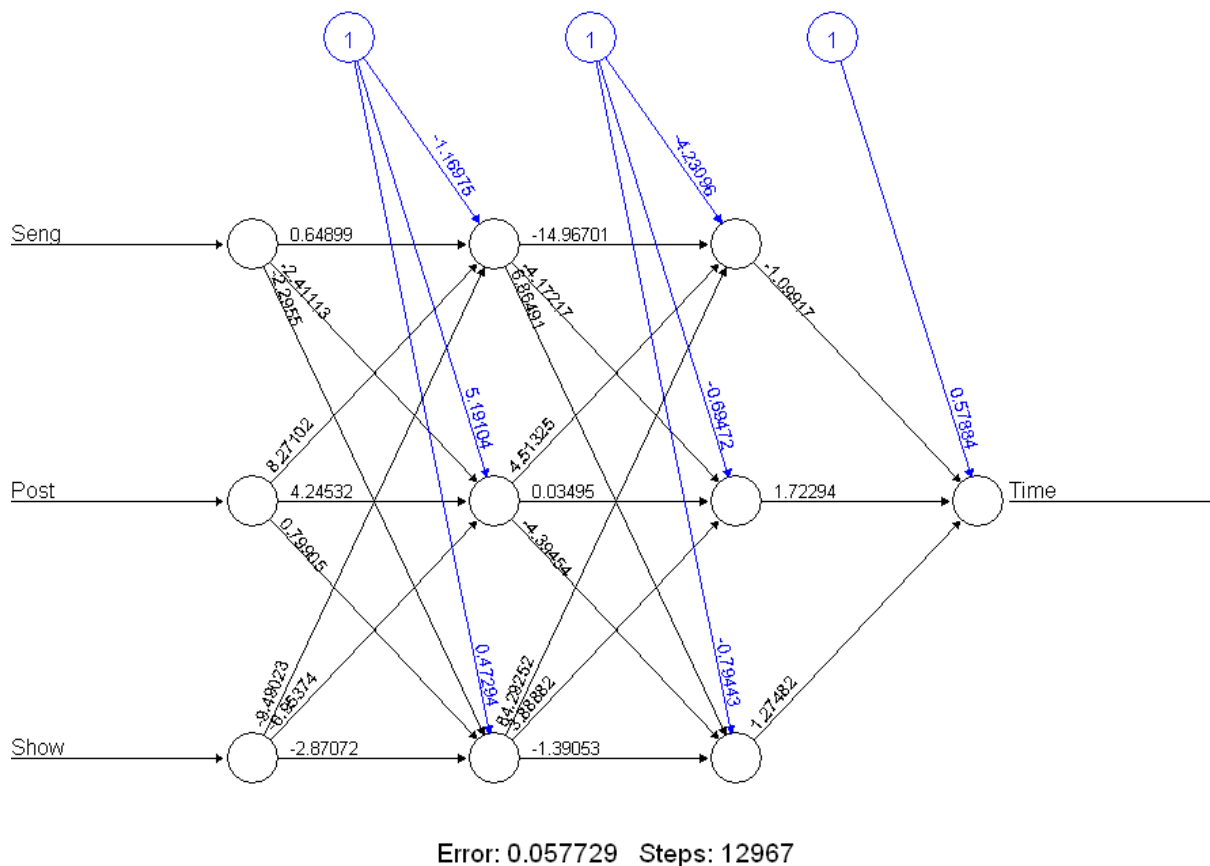


Рисунок 7 – Граф нейронної мережі моделі №3

```
> w2
```

	Seng	Post	Show	Traf	Conv	Time	Sum	res	error
7	20	12	526	1103	53	101	0.3094	101.42	0.0421
12	22	13	566	1050	65	109	1.2483	101.21	0.7794
13	23	13	552	1213	86	103	1.4275	98.44	0.4556
16	25	13	577	1046	72	106	1.0857	103.01	0.2985
19	21	11	596	1288	52	108	1.3764	103.65	0.4345
21	20	10	525	1137	97	110	2.1973	107.58	0.2424
22	20	12	580	1098	92	107	1.7114	103.77	0.3227
23	21	12	559	1190	99	109	2.2501	103.76	0.5242
27	23	12	516	1358	90	108	2.3148	103.43	0.4572

Рисунок 8 – Результати розрахунків за моделлю №3

Після запусків скрипта моделі №4 для різних параметрів кількості прихованих шарів та кількості нейронів у них з'ясовано, що найкращий результат (кореляція – 0,7829; середня абсолютна приведена помилка – 3,437) забезпечує персептрон з одним прихованим шаром та трьома нейронами у ньому (рис. 9, рис. 10). Але значення помилки та аналіз результатів свідчать, що модель №4 не може вважатися такою, що може мати значення.

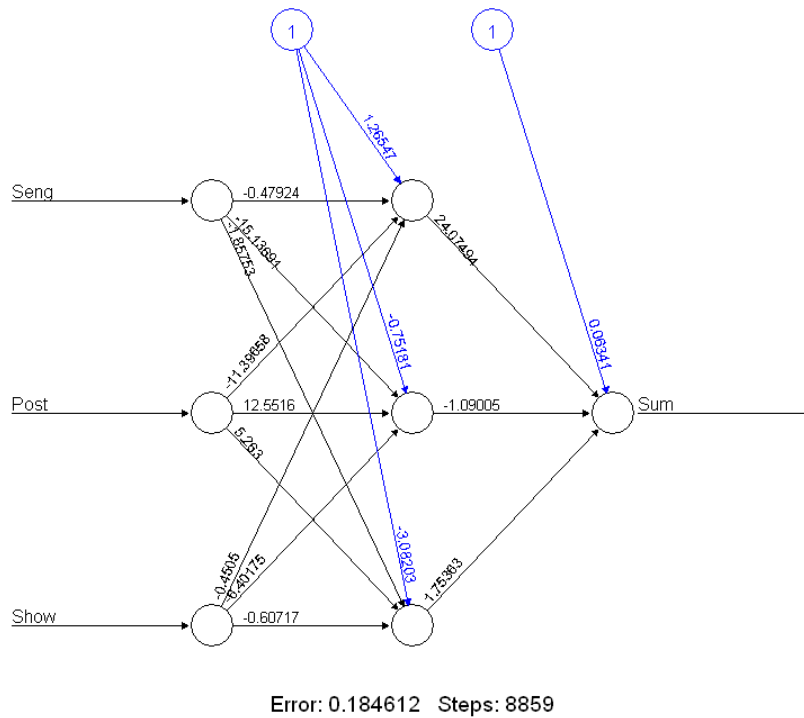


Рисунок 9 – Граф нейронної мережі моделі №4

```
> w<
  Seng Post Show Traf Conv Time Sum res error
1    25  15  540 1103   65  110 1.4594  2.641  0.54258
6    23  11  539 1250   59  102 0.8426 11.064  4.69281
10   21  11  599 1358   64  104 1.3731  9.861  3.89705
17   25  13  554 1430   51  102 1.0532  1.115  0.02821
18   25  13  538 1023   53  110 1.0417  1.179  0.06328
19   21  11  596 1288   52  108 1.3764  9.985  3.95225
21   20  10  525 1137   97  110 2.1973 40.542 17.60457
26   23  13  514 1252   51  105 0.9801  1.035  0.02504
28   26  12  530 1054   74  109 1.4442  1.717  0.12531
```

Рисунок 10 – Результати розрахунків за моделлю №4

Література

1. Федішин І. Б., Електронний бізнес та електронна комерція (опорний конспект лекцій). Тернопіль: ТНТУ імені Івана Пулюя, 2016. 155 с.
2. Краус К. М., Краус Н. М., Манжура О. В., Електронна комерція та Інтернет-торгівля: навчально-методичний посібник. Київ: Аграр Медіа Груп, 2021. 454 с.
3. Дrajниця С. А. Електронна комерція: навч. посібник. Львів: Новий світ-2000, 2013. 182 с.
4. Юдін О. М., Макарова М. В., Лавренюк Р. М., Системи електронної комерції: створення, просування і розвиток: монографія. Полтава: РВВ ПУЕТ, 2011. 201 с.
5. Кривінченко Д. Р., Мельников О. Ю. Задача розробки інформаційної системи для оцінки ефективності просування сайту. *Наукові досягнення та відкриття сучасної молоді [Електронний ресурс]: зб. матер. II Всеукр. наук. конф. студ. та молодих вчених (Луцьк, 31 трав. 2023 р.) / Держ. вищ. навч. заклад «Донецький національний технічний університет».* Луцьк : ДВНЗ «ДонНТУ», 2023. С. 31–36.
6. Мельников О. Ю. R – мова програмування та аналізу даних: навчальний посібник для здобувачів вищої освіти за спеціальностями «Системний аналіз» та «Інформаційні системи та технології». Краматорськ: ДДМА, 2023. 272 с.

**МАТЕРІАЛИ
ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ
КОНФЕРЕНЦІЇ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ
ТА МОЛОДИХ УЧЕНИХ**

**«СУЧАСНИЙ СТАН ДОСЛІДЖЕНЬ
В СФЕРІ ІТ»**

(19 квітня 2024 року)

Відповідальний за випуск

С.В. Пронін

В авторській редакції