

<https://doi.org/10.31891/2219-9365-2024-77-15>

УДК 004.738.5:519.68:004.45

ДИВАК Микола

Західноукраїнський національний університет

<https://orcid.org/0000-0002-9049-4993>

e-mail: mdy@wunu.edu.ua

МАНЖУЛА Володимир

Західноукраїнський національний університет

<https://orcid.org/0000-0001-5222-8443>

e-mail: volodymyr.manzhula@gmail.com

МЕЛЬНИК Андрій

Західноукраїнський національний університет

<https://orcid.org/0000-0001-7799-9877>

e-mail: melnyk.andriy@gmail.com

ЮШКО Андрій

Західноукраїнський національний університет

<https://orcid.org/0009-0003-6431-3479>

e-mail: a.yushko@wunu.edu.ua

АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ НА ОСНОВІ АНАЛІЗУ ІНТЕРВАЛЬНИХ ДАНИХ З ВИКОРИСТАННЯМ ХМАРНИХ ТЕХНОЛОГІЙ

В рамках цієї роботи запропоновано архітектуру програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. Особливостями запропонованої архітектури є імплементація підсистеми інтервального моделювання статичних систем в хмарній сервісно-орієнтованій архітектурі із врахуванням оптимізації обчислювальних схем за допомогою використання платформи Google Cloud Run, моделі розподілених обчислень MapReduce, безкоштовних програмно-інтерпретованих засобів та використання RESTful API на всіх етапах математичного моделювання.

Приведено ряд діаграм, які використовуються для ілюстрації особливостей запропонованої архітектури програмного забезпечення та описано специфіку їх імплементації в процесі практичної реалізації програмної системи для математичного моделювання.

На базі запропонованої архітектури реалізовано програмну систему для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. Одним із ключових компонентів даної системи є інтелектуалізована, консультативно-діагностична підсистема на базі ChatGPT, використання якої дозволило суттєво покращити практичне використання запропонованих рішень.

Ключові слова: архітектура програмного забезпечення, інтервальне моделювання, обчислювальна складність, хмарні технології, програмне середовище, API інтерфейс, ChatGPT.

DYVAK Mykola, MANZHULA Volodymyr,

MELNYK Andriy, YUSHKO Andriy

West Ukrainian National University

SOFTWARE ARCHITECTURE FOR MATHEMATICAL MODELING BASED ON ANALYSIS OF INTERVAL DATA USING CLOUD TECHNOLOGIES

As part of this work, a software architecture for mathematical modeling based on the analysis of interval data using cloud technologies is proposed. The features of the proposed architecture are the implementation of the subsystem of interval modeling of static systems in a cloud service-oriented architecture, taking into account the optimization of computing schemes using the Google Cloud Run platform, the MapReduce distributed computing model, free software-interpreted tools and the use of RESTful APIs at all stages of mathematical modeling.

A number of diagrams are given that are used to illustrate the features of the proposed software architecture, and the specifics of their implementation in the process of practical implementation of the software system for mathematical modeling are described.

Based on the proposed architecture, a software system for mathematical modeling based on the analysis of interval data using cloud technologies has been implemented. One of the key components of this system is an intellectualized, consultative and diagnostic subsystem based on ChatGPT, the use of which allowed to significantly improve the practical use of the proposed solutions.

Keywords: software architecture, interval modeling, computational complexity, cloud technologies, software environment, API interface, ChatGPT.

Постановка проблеми у загальному вигляді

та її зв'язок із важливими науковими чи практичними завданнями

Математичне моделювання є одним із ефективних інструментів не тільки для проведення досліджень, але і для розв'язування широкого кола прикладних задач, де необхідно приймати складні рішення, зокрема за допомогою інтелектуалізованих систем підтримки рішень та експертних систем [1]. Наближеність до користувача, зручність у застосуванні методів математичного моделювання реалізують за рахунок використання різноманітних програмних середовищ, які переважно інтегрують спектр математичних методів в єдину концепцію побудови інструментальних засобів, зрозумілу для користувачів – практиків. Прикладом таких середовищ є засоби: MATLAB, Mathematica або програмні бібліотеки Python, такі як NumPy і SciPy. Разом з тим, потреба застосування спеціалізованих методів, які розробляються рядом дослідників, наприклад методів аналізу інтервальних даних [2-4], вимагає додаткового опрацювання з метою розробки програмного інструментарію. Такий підхід є особливо актуальним, коли задачі стосуються інтеграції широкого кола засобів, наприклад, таких як штучний інтелект, забезпечення відкритості інструментального середовища, інтеграції даних у різних форматах представлення, застосування хмарних технологій тощо. В цьому напрямку проводиться ряд наукових досліджень, серед яких варто відзначити поєднання кількісних методів аналізу даних і моделювання з онтологічним підходом [5]. Проте, зазначений підхід, розроблений авторами праць [4,5] вимагає подальшого розвитку у напрямку інтеграції в єдине середовище ширшого кола методів ідентифікації моделей на основі аналізу даних, інтеграцію даних з хмарних середовищ, інтеграцію засобів штучного інтелекту, покращення якості користувацького інтерфейсу, а також використання хмарних середовищ для розподілених обчислень тощо. Методологічно, для розробки математичних моделей на загальному рівні найчастіше розглядають два підходи – дедуктивний та індуктивний, особливості яких проілюстровано на рисунку 1. При дедуктивному підході спочатку потрібно визначити загальну структуру моделі, а потім перевірити її відповідність експериментальним даним. Це ускладнює застосування цього підходу для побудови макромоделей систем підтримки прийняття рішень, які керуються моделями та даними [3-5].



Рис. 1. Підходи до математичного моделювання

Індуктивний підхід, навпаки, базується на експериментальному вивченні властивостей складного об'єкту з подальшою ідентифікацією математичної моделі. Моделі, побудовані в рамках цього підходу, є простими і їх часто використовують в інтелектуалізованих системах підтримки рішень та експертних системах.

Проте, для розробки таких моделей потрібно вирішувати складні оптимізаційні завдання ідентифікації. Особливо ці завдання ускладнюються, коли експериментальні дані подані у вигляді числових інтервалів, а вимоги до математичних моделей є достатньо жорсткими - гарантовані прогностичні властивості [2,3].

Виходячи з припущення, що точність математичної моделі повинна відповідати точності результатів експерименту, які подані у вигляді числових інтервалів і гарантовано включають спостережувані значення характеристик об'єкта, для створення таких моделей доцільно використовувати методи аналізу інтервальних даних. Самі моделі матимуть вигляд інтервальних рівнянь [4].

Хоча представлення такого класу моделей у вигляді інтервальних рівнянь має свої переваги для комп'ютерного моделювання, воно вимагає застосування складніших методів структурної та параметричної ідентифікації, у порівнянні із використанням стохастичних підходів [2,3]. Таким чином, проблема обчислювальної складності в задачах аналізу інтервальних даних та задачах ідентифікації математичних моделей на основі цих даних вимагає застосування певних технологічних рішень, які необхідно урахувати при розробці інструментального програмного середовища.

Одним із способів подолання проблеми обчислювальної складності є використання ефективних метаевристичних методів, які є більш ефективнішими у випадку великих обсягів даних або містять велику кількість ітераційних послідовностей. Цей процес може включати оптимізацію алгоритмів, використання паралельних обчислень або розробку спеціалізованих методів для конкретних типів задач (рисунок 2).

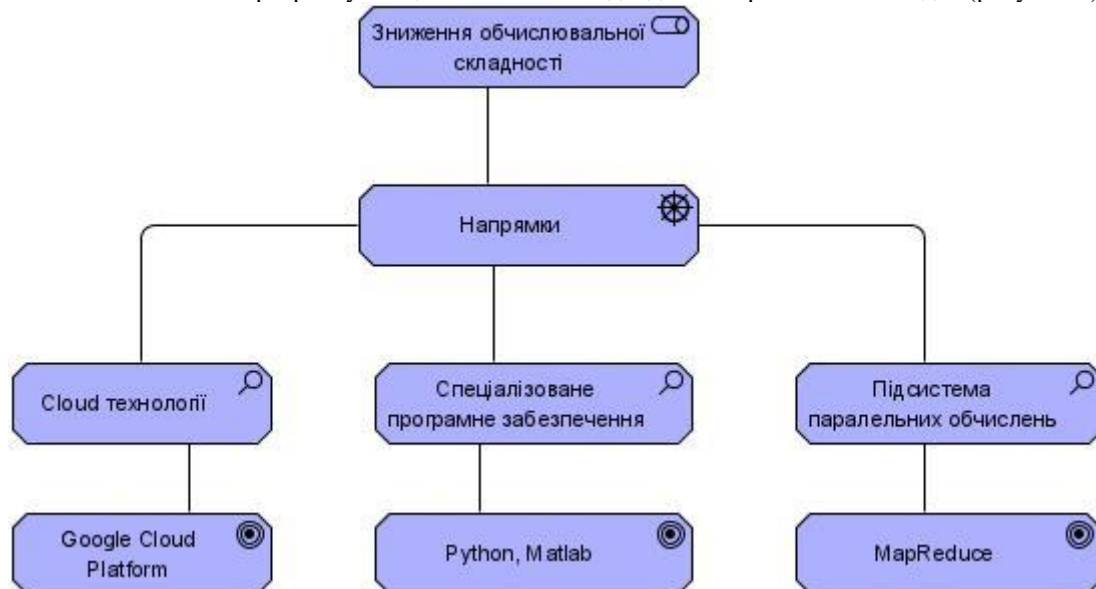


Рис. 2. Напрямки зниження обчислювальної складності з використанням інформаційних технологій

Як бачимо із рисунка 2, поряд із традиційними методами та технологіями вирішення проблеми обчислювальної складності, є використання хмарних сервісів та високопродуктивних обчислювальних систем. Саме цей аспект є предметом досліджень у статті. Проте, його урахування вимагає розробки архітектури програмної системи, яка інтегруватиме відповідні хмарні сервіси і забезпечуватиме вирішення проблеми високої обчислювальної складності в задачах розробки математичних моделей на основі аналізу інтервальних даних.

Використання хмарних сервісів для підтримки процесів математичного моделювання

Дослідження, що стосуються розробки систем математичного моделювання на основі хмарних технологій, стають все більш актуальними [6-9]. Це пов'язано насамперед з тим, що хмарні сервіси стають все більш доступними, досить масштабованими та сприяють зростанню ефективності обчислювальних систем. Ці системи дозволяють дослідникам, інженерам і аналітикам ефективно виконувати складне математичне моделювання, яке також включає аналіз даних та відповідні оптимізаційні процеси для задач ідентифікації моделей [10-12]. Основні компоненти структури програмної системи для математичного моделювання з використанням хмарних технологій наведено на рисунку 3. Розглянемо більш детально особливості побудови цієї системи.

Основною компонентою такої системи є хмарна інфраструктура, яка включає інфраструктуру, надану такими платформами, як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP) [8-12]. Зазначені платформи пропонують масштабовані обчислювальні ресурси, сховища даних та відповідні мережеві можливості, які можна динамічно розвивати відповідно до встановлених вимог.



Рис. 3. Загальна структура засобів реалізації програмної системи для математичного моделювання з використанням хмарних технологій

Необхідно також відзначити компоненту у вигляді спеціалізованого програмного забезпечення для моделювання, яку можна використовувати для реалізації відповідних обчислювальних процедур. Наприклад, такими компонентами можуть бути: MATLAB, Mathematica або програмні бібліотеки Python, такі як NumPy і SciPy [13-15]. Ці інструменти надають широкий спектр математичних функцій, відповідні процедури для обчислення розв'язків широко класу рівнянь та широкий спектр можливостей візуалізації результатів моделювання.

Як вже зазначалось, оскільки процес математичного моделювання супроводжується складними обчислювальними процедурами, то постає актуальне питання реалізації паралельних та розподілених обчислень. Хмарні технології дають можливість розпаралелювати обчислювальні завдання на кількох віртуальних машинах або контейнерах. Паралельні алгоритми та MapReduce можна використовувати для прискорення виконання складних обчислювальних етапів математичного моделювання, наприклад це стосується задач ідентифікації моделей на основі аналізу даних.

Особливо необхідно відзначити таку компоненту, як підсистема зберігання інформації, оскільки швидкість та варіативність доступу до даних є дуже важливими характеристиками для прискорення процесів математичного моделювання. Найпопулярнішими хмарними рішеннями для зберігання та управління даними є Amazon S3, Azure Blob Storage або Google Cloud Storage, які використовуються для зберігання вхідних даних, проміжних результатів і вихідних файлів, створених під час реалізації процедур математичного моделювання. Крім того, для керування структурованими даними можна використовувати такі хмарні бази даних, як Amazon RDS, Azure SQL Database або Google Cloud SQL.

Оркестровка робочого процесу реалізуватиметься у спосіб використання таких інструментів як Apache Airflow, AWS Step Functions або Azure Data Factory, які дозволяють автоматизувати та управляти виконанням складних робочих процесів основних етапів математичного моделювання. Ці інструменти дозволяють користувачам визначати, планувати та контролювати послідовність обчислювальних завдань із врахуванням критеріїв масштабованості та відмовостійкості.

Контейнеризація та мікросервіси: Технології контейнеризації, такі як Docker, і платформи керування контейнерами, як-от Kubernetes, зазвичай використовують для розгортання окремих програмних компонентів математичного моделювання як мікросервісів у хмарних середовищах. Такий підхід забезпечує модульність, масштабованість і портативність компонентів системи та дозволяє деяку варіативність реалізації з використанням різноманітних технологій.

Програмні системи математичного моделювання з використанням хмарних технологій для захисту даних і забезпечення конфіденційності, цілісності та доступності використовуватимуть засоби шифрування, контролю доступу та дотримання відповідних стандартів (наприклад, GDPR, HIPAA).

Хмарні інструменти моніторингу та оптимізації дають змогу зрозуміти продуктивність, використання ресурсів і вартість системи математичного моделювання та відповідних прикладних інформаційних систем. Для максимізації ефективності та мінімізації операційних витрат використовують такі методи, як автоматичне масштабування, правильний розмір екземпляра та алгоритми оптимізації.

Загалом інструментарій системи математичного моделювання, заснований на хмарних технологіях, забезпечує гнучкість, масштабованість і ефективність, надаючи можливість дослідникам і практикам вирішувати складні прикладні проблеми підтримки прийняття рішень. На рисунку 4 наведено приклад можливої загальної схеми комунікації системи моделювання із зовнішніми сервісами для застосування в медицині та екології.

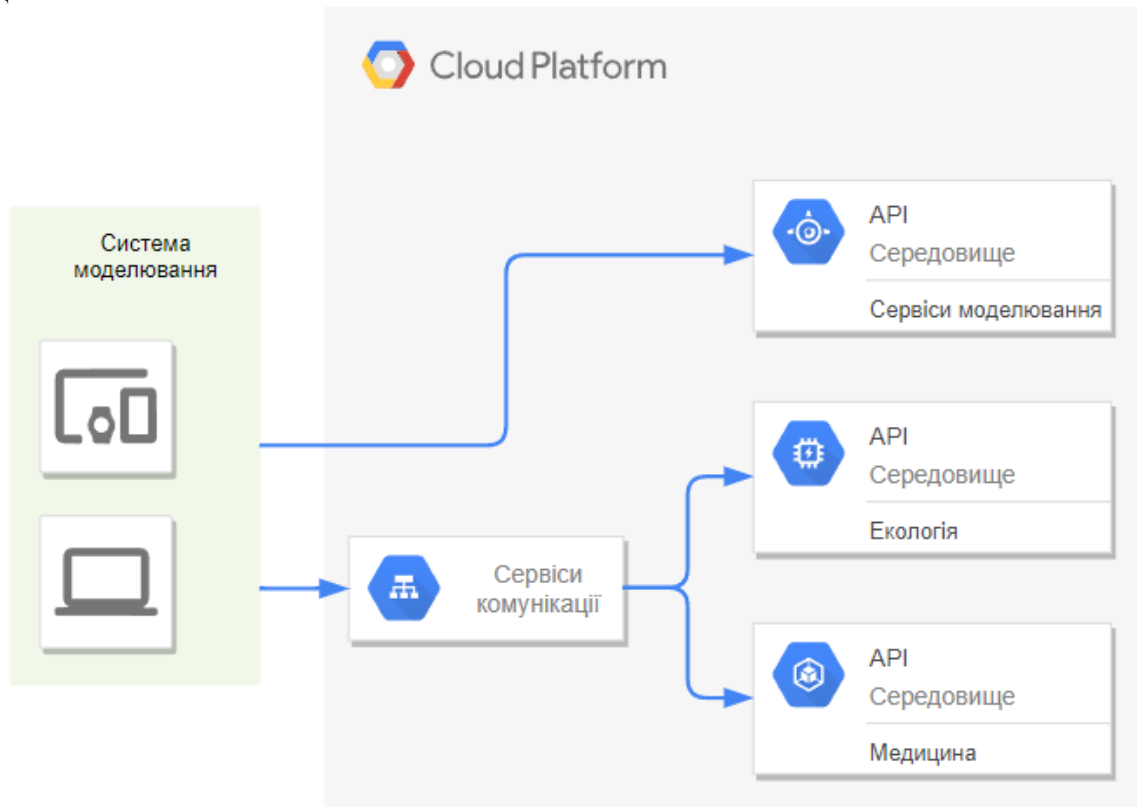


Рис. 4. Загальна схема комунікації системи моделювання із зовнішніми сервісами

Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій

Програмне забезпечення для математичного моделювання на основі аналізу інтервальних даних, яке направлене на побудову математичних моделей, які описують складні процеси у різних прикладних сферах повинне мати можливість інтеграції з іншими системами або обчислювальними середовищами, яка розширить можливості використання систем такого класу у складних проектах. Наприклад, можливість обміну даними з базами даних, іншими програмами моделювання або спеціалізованими середовищами для обробки даних дозволяє створювати більш комплексні моделі та забезпечує більш ефективну роботу з великим обсягом інформації. Така інтеграція також дозволяє автоматизувати частину обчислювального процесу і покращує зручність користування програмним забезпеченням. На рисунку 5 наведено архітектуру програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій.

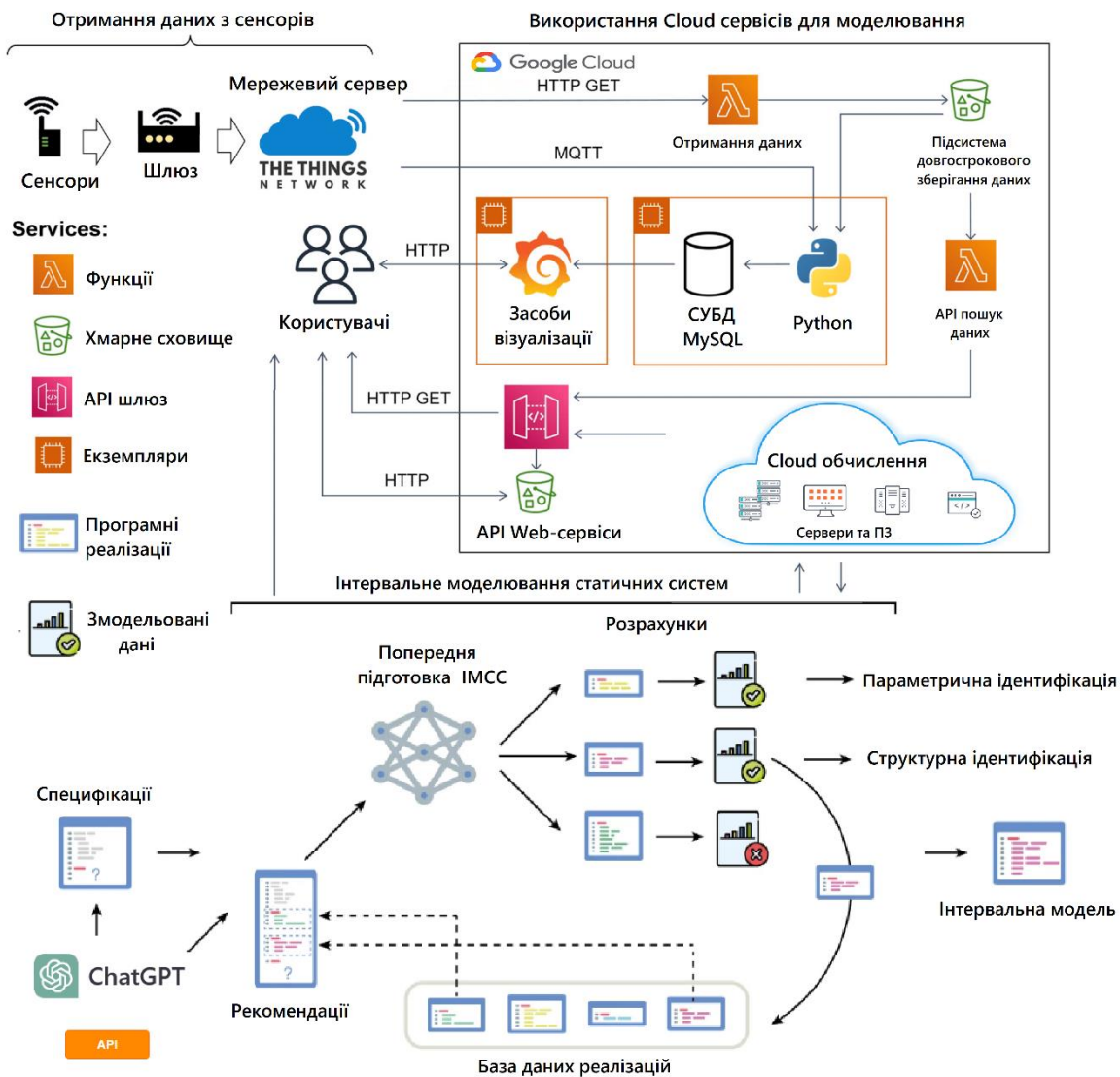


Рис. 5. Запропонована архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій

Враховуючи доступність та вартість програмних компонентів та орієнтування на безкоштовні засоби, основою системи є інструменти Google Cloud Platform, але ту саму архітектуру можна відтворити за допомогою еквівалентних сервісів інших хмарних постачальників. Розглянемо детальніше основні компоненти запропонованої архітектури, та опишемо їх основні взаємодії.

Платформа Google Cloud Run - це платформа для організації обчислень, яка ґрунтується на Knative і дозволяє запускати контейнери, доступ до яких здійснюється за допомогою HTTP-запитів. Вибір цієї платформи зумовлений тим, що користувачеві немає необхідності керувати компонентами хмарної інфраструктури, такими як обслуговування фізичних машин, оновлення програмного забезпечення, балансування навантаження або масштабування. Основна увага дослідників та розробників зосереджується суто на написанні програмного коду за допомогою будь-якої технології, використовуючи напрацьовані програмні бібліотеки з реалізованими математичними моделями або відповідні двійкові файли. Через відсутність потреби управління інфраструктурою, з'являється можливість сконцентруватися суто на написанні програмного коду, даючи можливість керованій обчислювальній платформі від Google здійснювати управління розподілом ресурсів при опрацюванні запитів.

Засоби RESTful API служать програмним інтерфейсом для швидкого завантаження даних із сенсорів, пошуку даних та моделей у сховищах, обміну даними з обчислюваними сервісами, програмно-інтерпретованим середовищем (Python, Matlab) та засобами візуалізації результатів моделювання. За допомогою служби Google API Gateway можна створити, публікувати та керувати API в хмарному середовищі Google Cloud. Також API використовується для читання, аналізу та запису даних із системи

довгострокового зберігання даних в хмарному середовищі. Щоб задокументувати засоби RESTful API і забезпечити легкий доступ до даних, які отримуються із різних сенсорів, пропонується статичний веб-сервіс за допомогою Swagger UI та його підключення CORS у службі API Gateway. Для нього додаємо спеціальний заголовок із маркером авторизації для контролю доступу. API описано відповідно до фреймворку Open API 3.0 і збережено як файл json, завантажений у змінну специфікації в коді JavaScript, як основу для формування підсистеми документації. Щоб завантажити дані за допомогою API, користувач повинен ввести дійсний маркер авторизації, щоб отримати необхідний доступ. Використовуючи API, можна запитувати набори даних для здійснення процедури моделювання з доступних сенсорів, так і з бази даних експериментальних досліджень. На рисунку 6 представлено схему організації взаємодії системи моделювання з використанням API інтерфейсів.

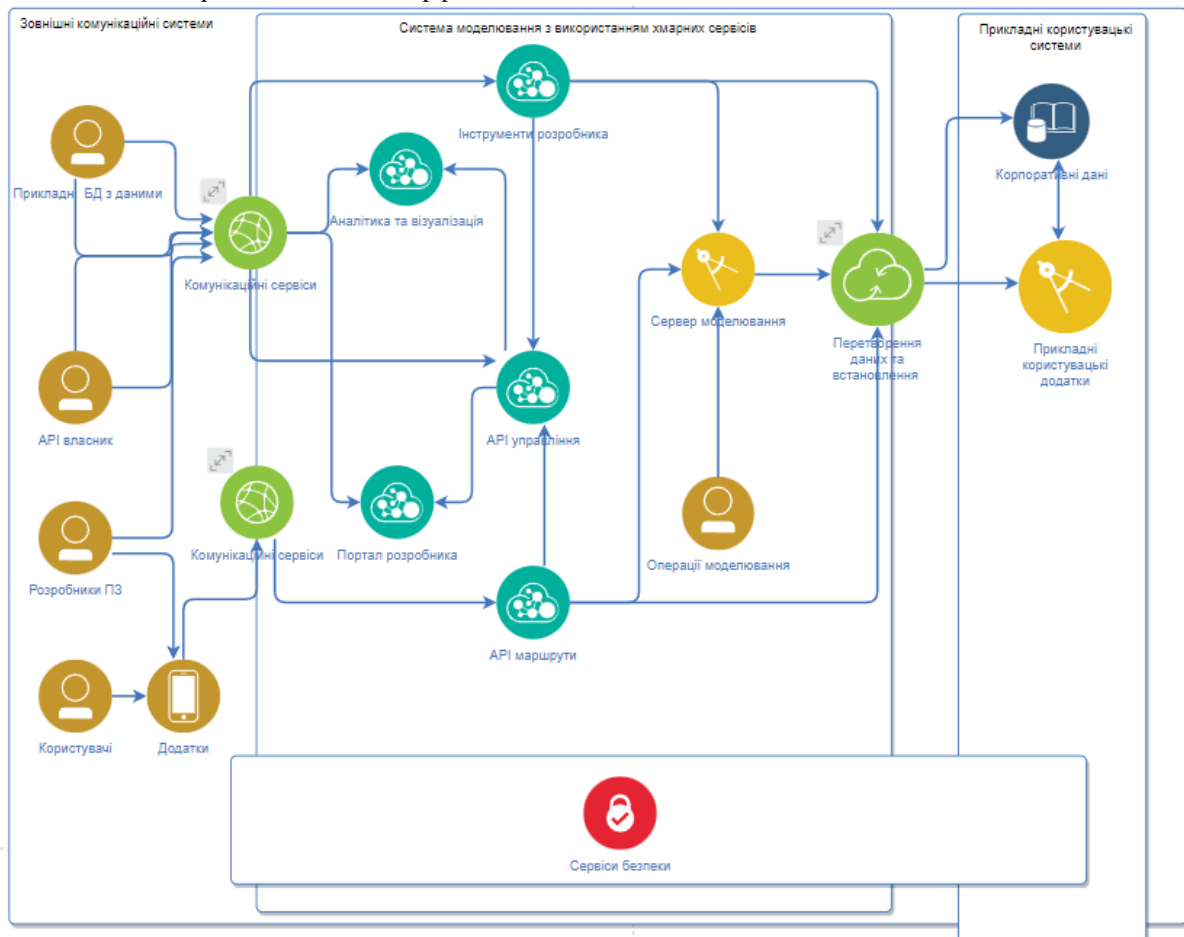


Рис. 6. Схема організації взаємодії системи моделювання з використанням API інтерфейсів

За головну компоненту, призначення якої є зберігання напрацьованих результатів моделювання є обрано MySQL як просте рішення з широкими засобами інформаційно-комунікаційної підтримки. Екземпляр MySQL був розгорнутий в хмарі через Bitnami, який надає попередньо налаштований образ віртуальної машини, готовий для завантаження в екземпляр Google Cloud Platform.

Підсистема візуалізації результатів математичного моделювання реалізована з використанням таких засобів:

- Matplotlib - одна з найпопулярніших бібліотек Python для створення двовимірних графіків та діаграм, що дозволяє створювати різноманітні типи графіків, такі як лінійні графіки, стовпчасті діаграми, точкові графіки тощо;
- Seaborn - бібліотека Python заснована на Matplotlib і надає вищий рівень абстракції для створення інформативніших графіків та дозволяє швидко створювати стилізовані графіки для аналізу даних;
- Plotly - бібліотека Python для створення інтерактивних візуалізацій, яка дозволяє створювати різноманітні типи графіків та діаграм, які дозволяють реалізовувати візуальну online взаємодію з користувачами;

- Vokeh - бібліотека Python, яку використано для створення інтерактивних візуалізацій, яка дозволяє створювати високоякісні візуалізації та надає інструменти для взаємодії з користувачем через веб-браузер.

Особливо важливою компонентою запропонованої архітектури є підсистема, яка описує специфіку інтервального моделювання складних систем. Процес математичного моделювання на основі інтервальних даних є складним і вимагає виконання деякої послідовності кроків, яка уточнюється в даній ситуації специфікою використання хмарних сервісів. Розглянемо ці кроки більш детально, але із врахуванням згаданої специфіки, яку в загальному представлено на рисунку 7.

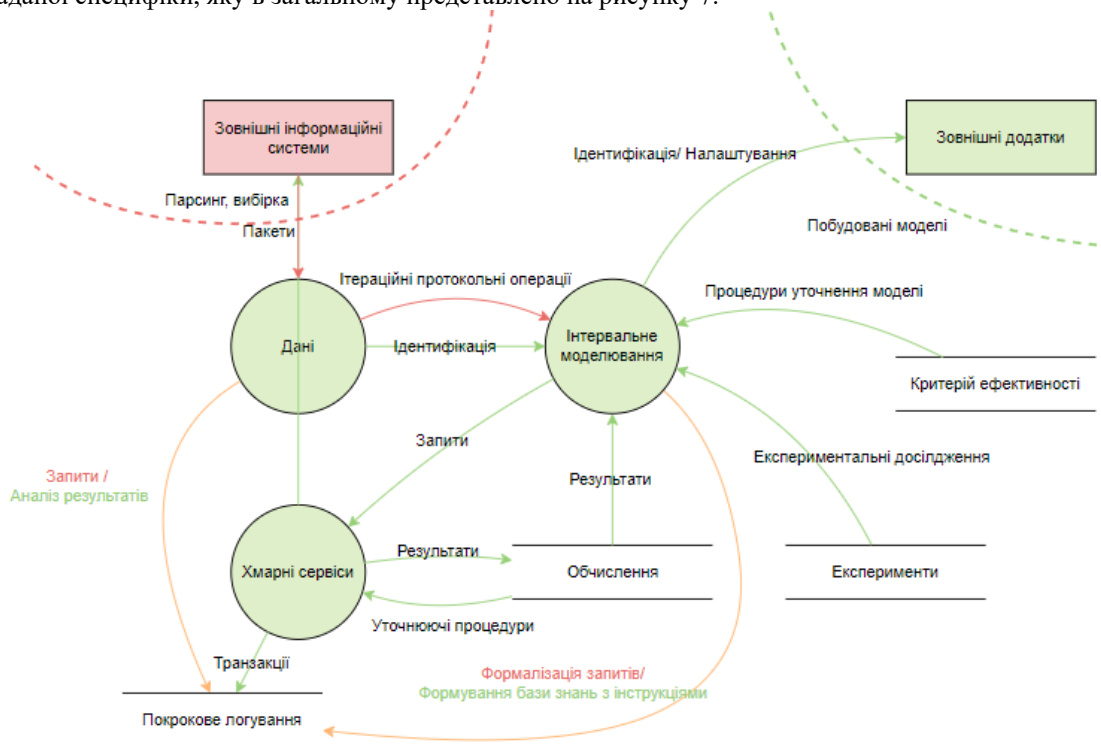


Рис. 7. Схема побудова інтервальної моделі з використанням хмарних сервісів

Процес побудови інтервальної моделі включає наведені нижче кроки.

1) Збір даних з сенсорів та/або напрацьованої бази експериментальних досліджень. Математична модель ґрунтується на даних, тому важливо зібрати і аналізувати відповідні дані, які характеризують поведінку системи. Цей етап є дуже важливим в рамках інтервального підходу, а використання API-інтерфейсів дозволяє пришвидшити отримання та синхронізувати дані між системою моделювання та прикладними користувацькими системами.

2) Вибір структури математичної моделі. В нашому випадку для вибору структури можуть бути використані математичні методи структурної ідентифікації. Проте для розв'язування цієї задачі необхідним є виконання низки умов. Перш за все це вибір критерію ефективності при побудові моделі. По-друге формування набору структурних елементів. Переважно це здійснюємо використовуючи фізичні міркування. Також можливо отримання структури моделі із використання онтологічного підходу [5].

3) Параметрична ідентифікація математичної моделі. Зазвичай її виконують як певні етапи структурної ідентифікації.

4) Перевірка та аналіз результатів. Оцінюють отримані результати, порівнюючи їх з реальними даними або іншими моделями.

5) Верифікація моделі. Якщо необхідно, модель перевіряється на точність та адекватність. Така перевірка здійснюється на основі використання програмно-інтерпретованого експертного середовища;

6) Використання та інтерпретація. Отриману модель можна використовувати для прогнозування, оптимізації або прийняття рішень у відповідній предметній області. Зазначений етап графічно проілюстровано на рисунку 8. Цю процедуру достатньо просто виконувати в рамках запропонованої архітектури, оскільки реалізовано цілу підсистему для обміну даними з прикладними системами.

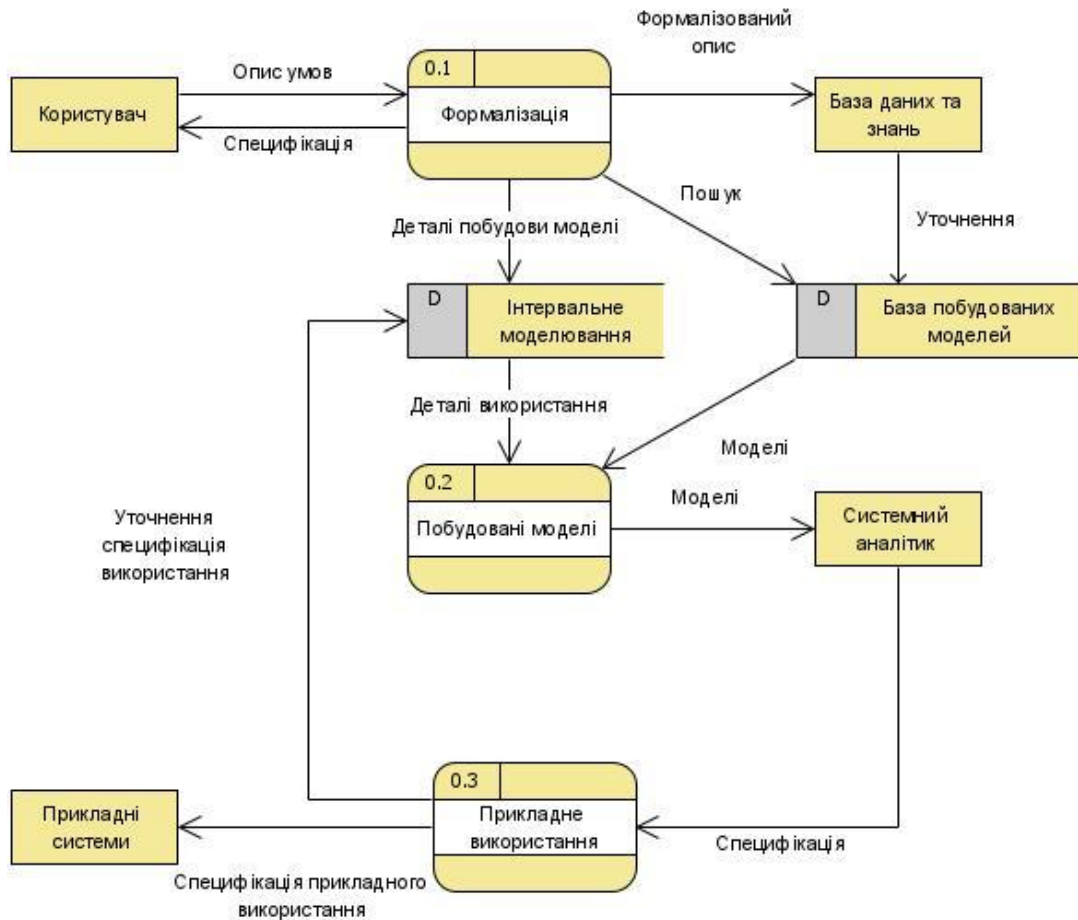


Рис. 8. Схема застосування розроблених інтервальних моделей

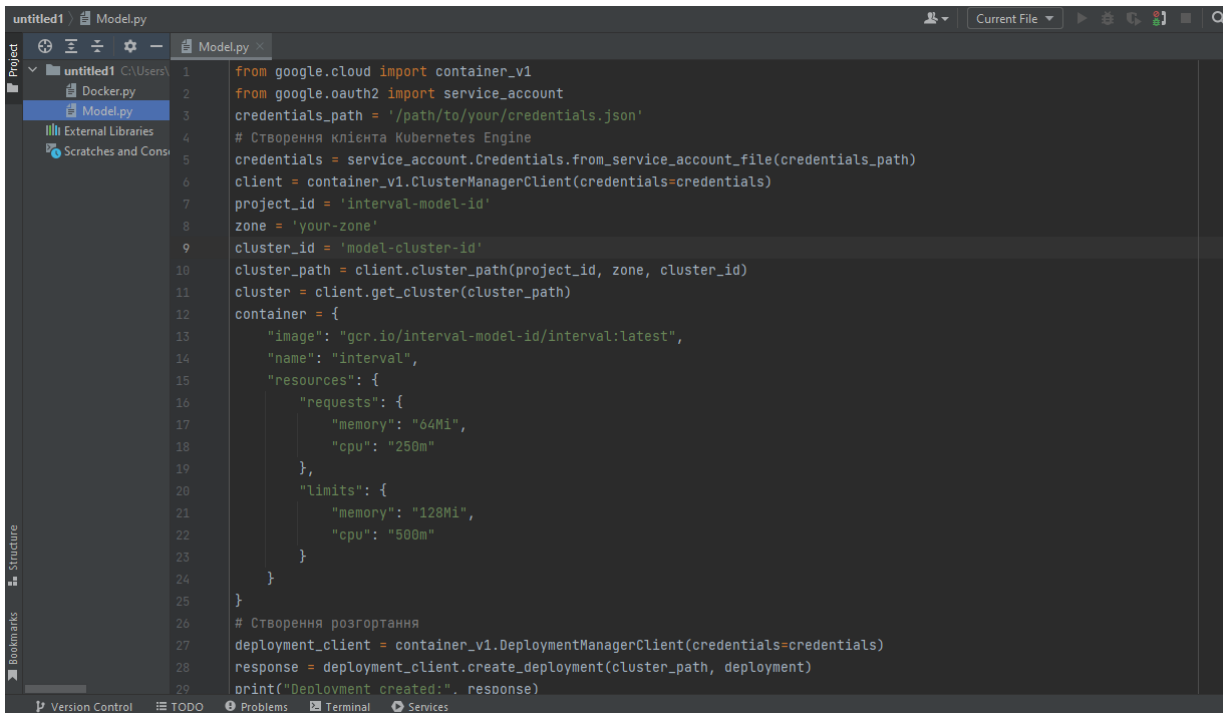
Одною із компонент запропонованої архітектури є реалізація інтелектуалізованої, консультативно-діагностичної підсистеми надання рекомендацій щодо процедур моделювання, процесів формалізованого опису моделей та їх практичного використання. Базою цієї компоненти використано ChatGPT, а його інтеграція реалізована з використанням відповідного API. ChatGPT API — це інструмент, доступний мовами програмування, що дозволяє інтегрувати функціональність ChatGPT у програмне забезпечення. Засоби Chat GPT OpenAI надають свій API через хмару, що дозволило отримати доступ до нього з будь-якого пристрою. Після розгортання API засоби моделювання отримали набір зручних інструментів, які забезпечили доступ до його використання як для спеціалістів в галузі математичного моделювання, так і для спеціалістами з прикладних галузей.

Запропонована архітектура програмної системи для математичного моделювання на основі хмарних сервісів дозволить підвищити ефективність при розробці математичних моделей на основі аналізу інтервальних даних та їх практичного використання. Архітектура є відкритою, що дозволяє легко розширювати та модифікувати модель, додаючи або змінюючи окремі складові без значних змін у концепції її побудови.

Використання швидких та ефективних метаевристичних алгоритмів для обробки даних дозволяє швидше отримувати результати та знижує обчислювальну складність. Інструменти візуалізації допомагають інтерпретувати та представляти дані у зрозумілій формі, що сприяє кращому розумінню процесів та характеристик об'єкта, які відображає модель. Архітектура побудована з урахуванням можливості масштабування, що дозволяє працювати з великими обсягами даних та розширювати обсяг моделювання за зміною потреб. Повторне використання напрацьованої бази моделей та експериментальних досліджень дозволяє адаптуватися до різних умов та задач, сприяючи її використанню в різних галузях та сценаріях. Запропонована архітектура при її реалізації забезпечить підвищення точності та адекватності математичних моделей, а також зменшенню часу, необхідного для їх розробки та аналізу.

Програмна реалізація системи для математичного моделювання на основі аналізу інтервальних даних

В рамках імплементації запропонованої архітектури реалізовано систему CloudInterval. Основною компонентою системи є хмарна інфраструктура, яка реалізована з використанням Google Cloud Platform. Для організації обчислень використано платформу Google Cloud Run, яка ґрунтується на Knative і дозволяє запускати контейнери, доступ до яких здійснюється за допомогою HTTP-запитів. Для реалізації обчислювальних процедур використано Python, зокрема, такі бібліотеки як NumPy і SciPy (рисунок 9).



```
1 from google.cloud import container_v1
2 from google.oauth2 import service_account
3 credentials_path = '/path/to/your/credentials.json'
4 # Створення клієнта Kubernetes Engine
5 credentials = service_account.Credentials.from_service_account_file(credentials_path)
6 client = container_v1.ClusterManagerClient(credentials=credentials)
7 project_id = 'interval-model-id'
8 zone = 'your-zone'
9 cluster_id = 'model-cluster-id'
10 cluster_path = client.cluster_path(project_id, zone, cluster_id)
11 cluster = client.get_cluster(cluster_path)
12 container = {
13     "image": "gcr.io/interval-model-id/interval:latest",
14     "name": "interval",
15     "resources": {
16         "requests": {
17             "memory": "64Mi",
18             "cpu": "250m"
19         },
20         "limits": {
21             "memory": "128Mi",
22             "cpu": "500m"
23         }
24     }
25 }
26 # Створення розгортання
27 deployment_client = container_v1.DeploymentManagerClient(credentials=credentials)
28 response = deployment_client.create_deployment(cluster_path, deployment)
29 print("Deployment created:", response)
```

Рис. 9. Використання Python для програмної реалізації інтервальних моделей

На рисунку 10 наведено діаграму послідовності програмної реалізації процесів побудови інтервальної математичної моделі. Як було зазначено вище, процеси математичного моделювання супроводжуються складними обчислювальними процедурами, які мають високу обчислювальну складність.

Для прискорення виконання складних обчислювальних етапів математичного моделювання використано модель розподілених обчислень від Google - MapReduce.

Для зберігання та управління даними в хмарі використано Google Cloud Storage, а процедури маніпуляції даними реалізовано з використанням Google Cloud SQL. В якості підсистеми локального зберігання результатів моделювання використано СУБД MySQL.

Технології контейнеризації, такі як Docker, і платформи керування контейнерами, як-от Kubernetes, використано для розгортання окремих програмних компонентів математичного моделювання як мікросервісів у хмарних середовищах. Керування контейнерами та образами в Docker мало чим відрізняється від керування програмами в терміналі Linux. Принцип той самий – вводимо текст і отримуємо результат. Спочатку треба вказати ключове слово Docker, потім вказати команду, яку треба виконати, а потім об'єкт застосування команди, аргументи, опції та інші додатки.

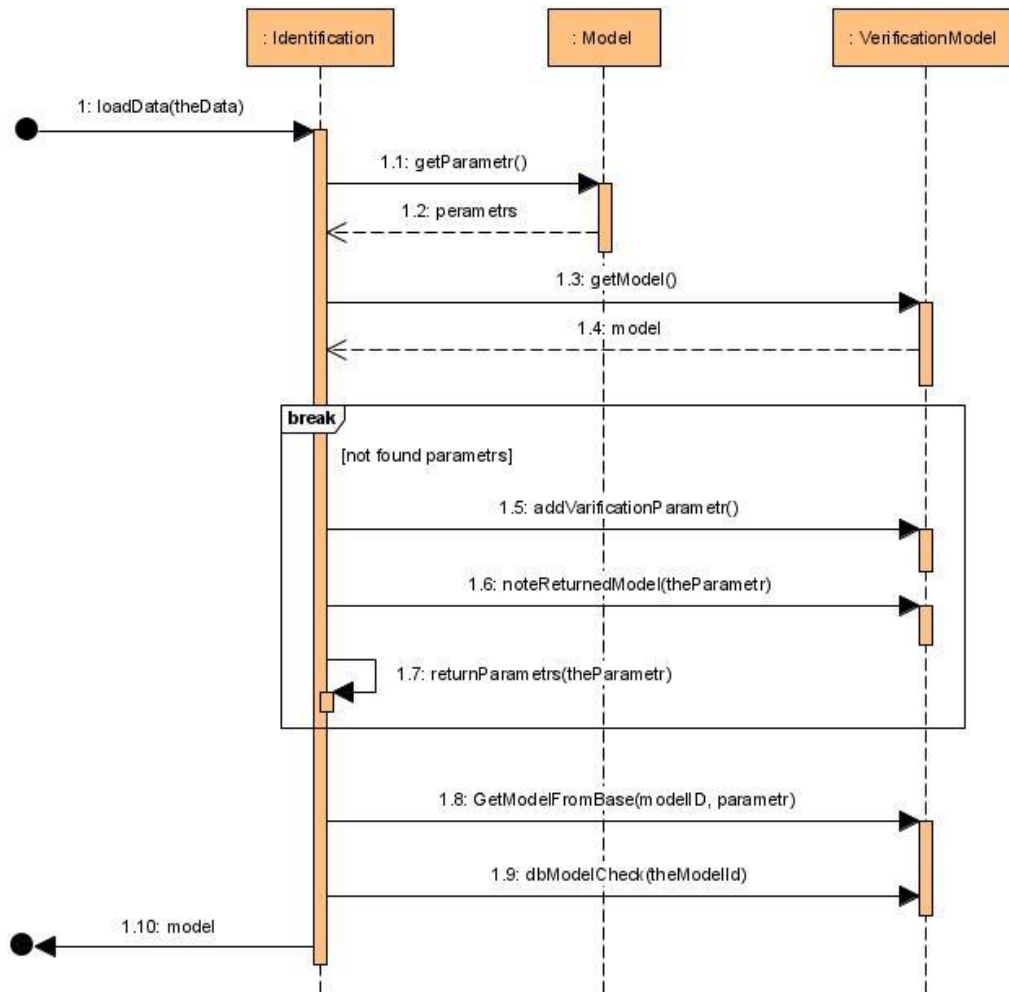


Рис.10. Діаграма послідовності процесів побудови математичної моделі

На рисунку 11 представлено скріншот, який відображає процедуру налаштування Docker, та відповідні засоби для виконання процедур.

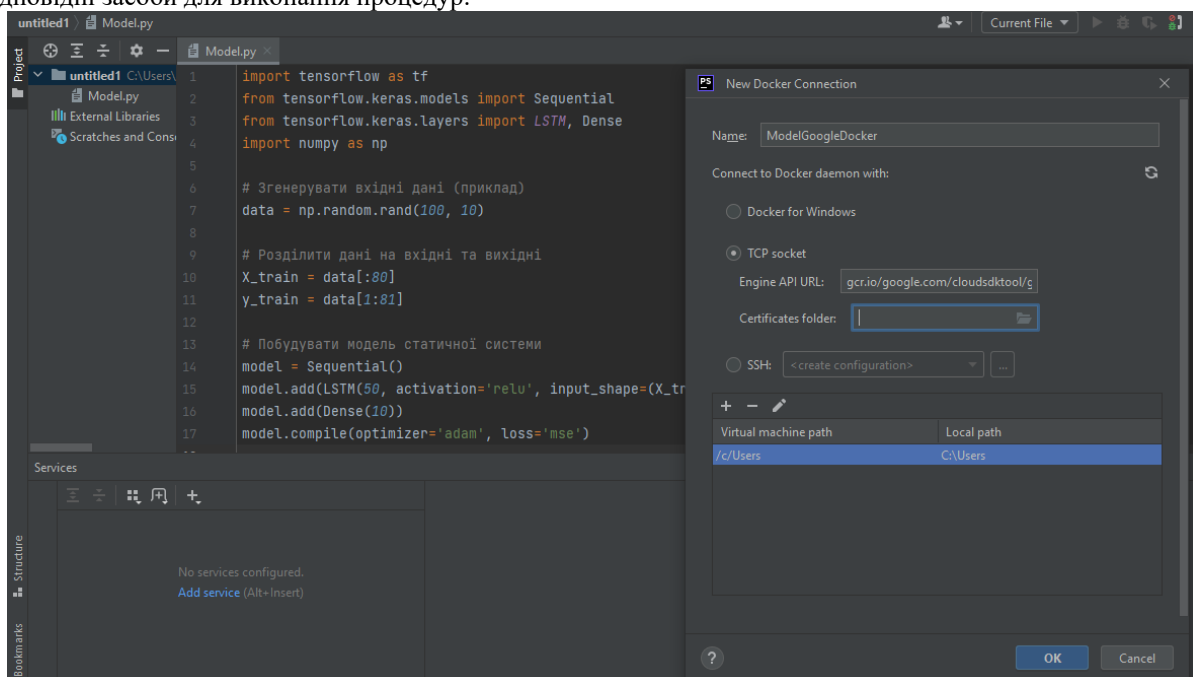


Рис.11. Екранна форма для відображення процедури налаштування для образу Docker

Приклади реалізації процесів моделювання із використанням хмарного середовища

Експериментальні дослідження програмної системи для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій були направлені на дослідження ефективності та особливостей запропонованого підходу. Розглянемо їх більш детально.

На рисунку 12 представлено приклад екранної форми в процесі реалізації моделі оцінки рівня забруднення шкідливими викидами автотранспорту в середовищі CloudInterval.

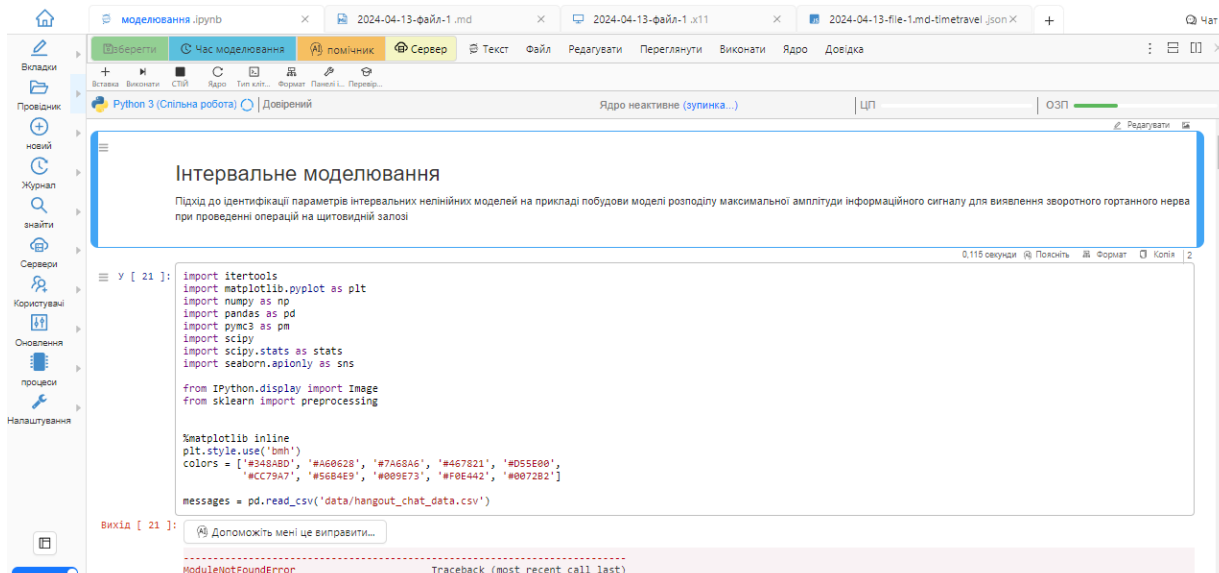


Рис. 12. Екранна форма для ілюстрації функціонування підсистеми інтервального моделювання в системі CloudInterval

Проведено дослідження можливостей оптимізації швидкодії та ефективності програмної системи для обробки великих обсягів інтервальних даних в хмарному середовищі. В результаті таких досліджень отримано показники зниження обчислювальної складності.

Визначення сфер застосування програмної системи і її можливостей для вирішення конкретних завдань та проблем підтверджено шляхом використання напрацьованих результатів для прикладних галузей.

На рисунку 13 представлено приклад візуалізації результатів математичного моделювання візуалізації розподілу максимальної амплітуди інформаційного сигналу на основі розробленої моделі для виявлення зворотного гортанного нерва при проведенні операцій на щитовидній залозі [4].

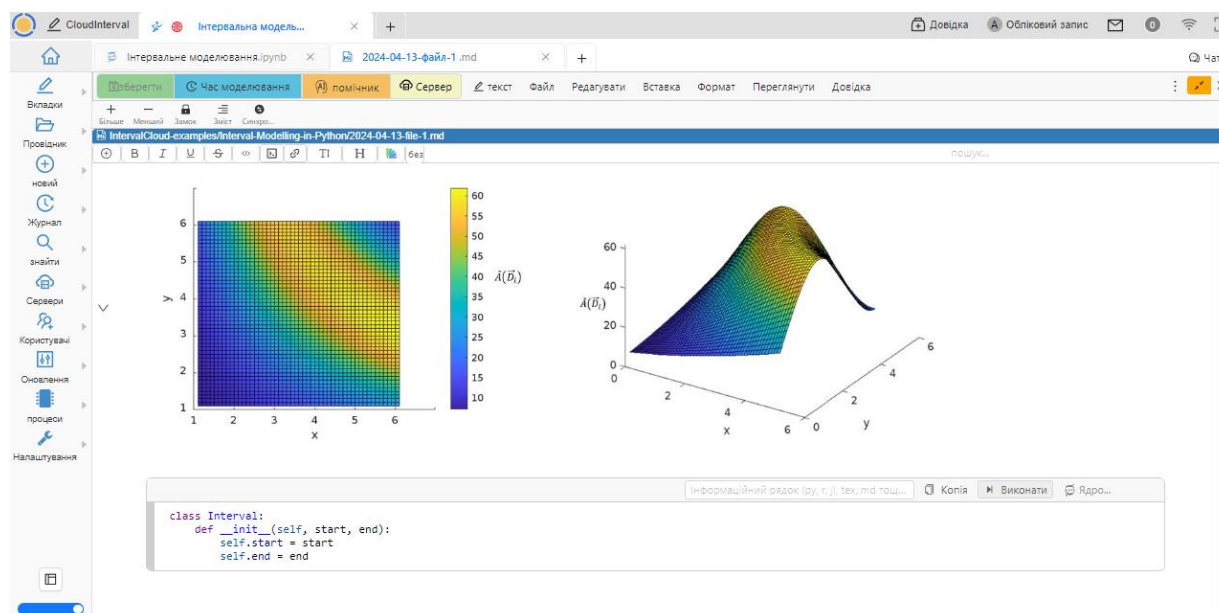


Рис. 13. Екранна форма для ілюстрації процесу візуалізації результатів математичного моделювання в системі CloudInterval

На рисунку 14 наведено приклад використання інтелектуалізованої, консультативно-діагностичної підсистеми на базі ChatGPT. Використання ChatGPT API сприяє швидкому вивченню основних можливостей системи та покращенню процесів розуміння процедур математичного моделювання.

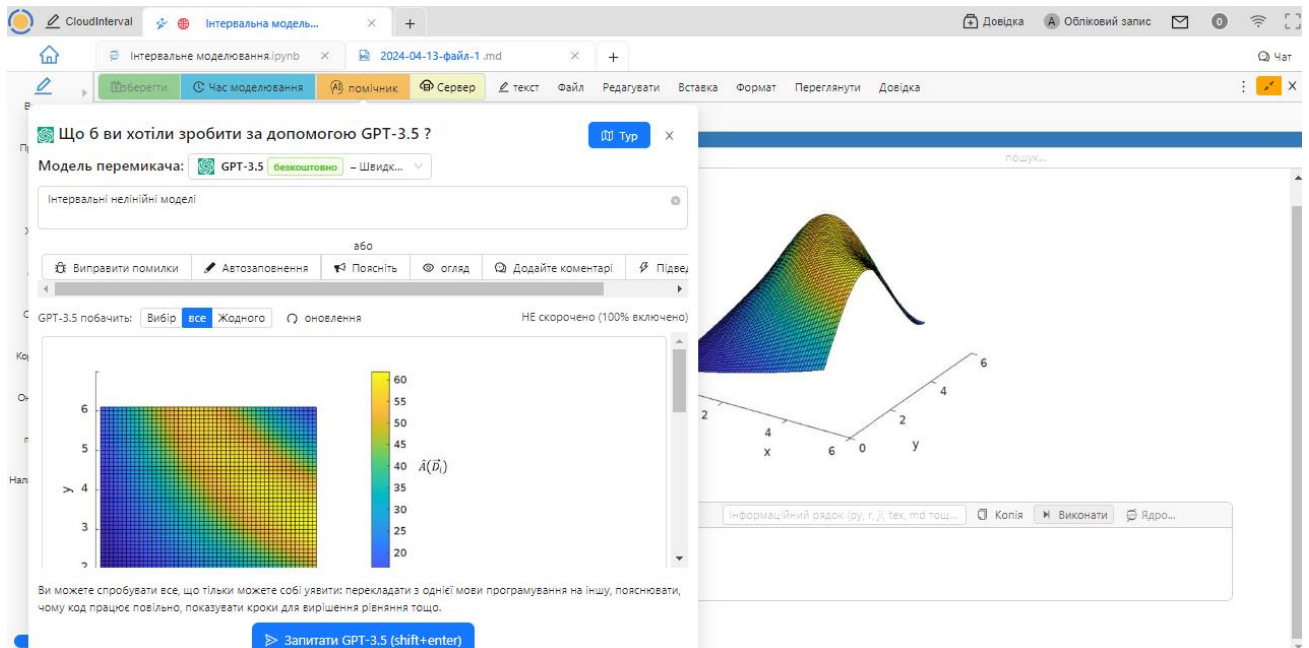


Рис. 14. Реалізація консультативно-діагностичної підсистеми на базі ChatGPT в системі CloudInterval

Проведені дослідження підтвердили можливості використання розробленої системи для математичного моделювання на основі методів аналізу інтервальних даних та хмарних технологій в прикладних галузях та сприяють розвитку нових інструментів та методів для аналізу та прогнозування.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

В роботі запропоновано та обґрунтовано архітектуру програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. В результаті проведених досліджень отримано такі результати:

1) Встановлено, що потреби наближеності до користувача, забезпечення зручності у застосуванні методів математичного моделювання на основі інтервальних даних вимагають розробки спеціалізованих програмних середовищ, які інтегрують спектр математичних методів в єдину концепцію побудови інструментальних засобів, зрозумілу для користувачів –практиків. Такий підхід можливо реалізувати із використанням сервісно-орієнтованої архітектури із урахуванням оптимізації обчислювальних процесів за допомогою застосування платформи Google Cloud Run, моделі розподілених обчислень MapReduce, безкоштовних програмно-інтерпретованих засобів та використання RESTful API.

2) Як базову методологію для математичного моделювання за цих умов обрано індуктивний підхід, суть якого полягає у налаштуванні математичної моделі під конкретні дані, представлені в інтервальному вигляді, які описують характеристики об'єкта за певних умов;

3) Вперше запропоновано та обґрунтовано архітектуру програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. Особливостями запропонованої архітектури є імплементація підсистеми інтервального моделювання статичних систем в хмарній сервісно-орієнтованій архітектурі із врахуванням оптимізації обчислювальних схем за допомогою використання платформи Google Cloud Run, моделі розподілених обчислень MapReduce, безкоштовних програмно-інтерпретованих засобів та використання RESTful API на всіх етапах математичного моделювання

4) На базі запропонованої архітектури реалізовано програмну систему для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. Одним із ключових компонентів даної системи є інтелектуалізована, консультативно-діагностична підсистема на базі ChatGPT, використання якої дозволить покращити практичне використання запропонованих рішень.

Література

1. О.Г.Івахненко. Індуктивні методи самоорганізації моделей складних систем - Київ: Наук.думка, 1982.
2. Дивак, М. П. Ідентифікація дискретних моделей систем з розподіленими параметрами на основі аналізу інтервальних даних [Текст] : монографія / М. П. Дивак, Н. П. Порплиця, Т. М. Дивак. – Тернопіль: Економічна думка ТНЕУ, 2018. – 220 с.
3. Дивак М. П. Задачі математичного моделювання статичних систем з інтервальними даними: монографія / М. П. Дивак. - Т. : Економ. думка ТНЕУ, 2011. - 215 с.
4. Дивак М. П. Прикладні задачі структурної та параметричної ідентифікації інтервальних моделей складних об'єктів [Електронний ресурс] : монографія / М. П. Дивак, А. В. Пукас, Н. П. Парплиця, А. М. Мельник. - Тернопіль : Університетська думка, 2021. - 212 с.
5. Mykola Dyvak, Andriy Melnyk, Artur Rot, Marcin Hernes, Andriy Pukas, "Ontology of mathematical modelling based on interval data", Complexity, vol. 2022, Article ID 8062969, 24 pages, 2022
6. Calderón-Gómez, Huriviades, Luis Mendoza-Pittí, Miguel Vargas-Lombardo, José Manuel Gómez-Pulido, Diego Rodríguez-Puyol, Gloria Sención, and María-Luz Polo-Luque. 2021. "Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment" Applied Sciences 11, no. 10: 4350. <https://doi.org/10.3390/app11104350>
7. Da Silva, M.A.P.; Times, V.C.; De Araujo, A.M.C.; Da Silva, P.C. A Microservice-Based Approach for Increasing Software Reusability in Health Applications. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
8. Leal Sobral, Victor Ariel, Jacob Nelson, Loza Asmare, Abdullah Mahmood, Glen Mitchell, Kwadwo Tenkorang, Conor Todd, Bradford Campbell, and Jonathan L. Goodall. 2023. "A Cloud-Based Data Storage and Visualization Tool for Smart City IoT: Flood Warning as an Example Application" Smart Cities 6, no. 3: 1416-1434. <https://doi.org/10.3390/smartcities6030068>
9. Barthelemy, J.; Amirghasemi, M.; Arshad, B.; Fay, C.; Forehead, H.; Hutchison, N.; Iqbal, U.; Li, Y.; Qian, Y.; Perez, P. Problem-Driven and Technology-Enabled Solutions for Safer Communities: The case of stormwater management in the Illawarra-Shoalhaven region (NSW, Australia). In Handbook of Smart Cities; Springer: Berlin, Germany, 2020; pp. 1–28.
10. Shanmuga Sundaram, J.P.; Du, W.; Zhao, Z. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. IEEE Commun. Surv. Tutor. 2020, 22, 371–388.
11. Luft, Angela, Nils Luft, and Kristian Arntz. 2023. "A Basic Description Logic for Service-Oriented Architecture in Factory Planning and Operational Control in the Age of Industry 4.0" Applied Sciences 13, no. 13: 7610. <https://doi.org/10.3390/app13137610>
12. Perri, Damiano, Marco Simonetti, and Osvaldo Gervasi. 2022. "Deploying Efficiently Modern Applications on Cloud" Electronics 11, no. 3: 450. <https://doi.org/10.3390/electronics11030450>
13. Betarte, G.; Giménez, E.; Martínez, R.; Álvaro, P. Machine learning-assisted virtual patching of web applications. arXiv 2018, arXiv:cs.CR/1803.05529.
14. Schnieder, Maren, and Sheryl Williams. 2023. "Educational Mobile Apps for Programming in Python: Review and Analysis" Education Sciences 13, no. 1: 66. <https://doi.org/10.3390/educsci13010066>
15. Guedj, Benjamin, and Bhargav Srinivasa Desikan. 2020. "Kernel-Based Ensemble Learning in Python" Information 11, no. 2: 63. <https://doi.org/10.3390/info11020063>

References

1. O.H.Ivakhnenko. Induktyvni metody samoorhanizatsii modelei skladnykh system - Kyiv: Nauk.dumka, 1982.
2. Dyvak, M. P. Identyfikatsiia dyskretnykh modelei system z rozpodilenyimi parametramy na osnovi analizu intervalnykh danykh [Tekst] : monohrafiia / M. P. Dyvak, N. P. Porplytsia, T. M. Dyvak. – Ternopil: Ekonomichna dumka TNEU, 2018. – 220 s.
3. Dyvak M. P. Zadachi matematychnoho modeliuвання statychnykh system z intervalnyimi danymi: monohrafiia / M. P. Dyvak. - T. : Ekonom. dumka TNEU, 2011. - 215 c.
4. Dyvak M. P. Prykladni zadachi strukturnoi ta parametrychnoi identyfikatsii intervalnykh modelei skladnykh ob'iektiv [Elektronnyi resurs] : monohrafiia / M. P. Dyvak, A. V. Pukas, N. P. Parplytsia, A. M. Melnyk. - Ternopil : Universytetska dumka, 2021. - 212 s.
5. Mykola Dyvak, Andriy Melnyk, Artur Rot, Marcin Hernes, Andriy Pukas, "Ontology of mathematical modelling based on interval data", Complexity, vol. 2022, Article ID 8062969, 24 pages, 2022
6. Calderón-Gómez, Huriviades, Luis Mendoza-Pittí, Miguel Vargas-Lombardo, José Manuel Gómez-Pulido, Diego Rodríguez-Puyol, Gloria Sención, and María-Luz Polo-Luque. 2021. "Evaluating Service-Oriented and Microservice Architecture Patterns to Deploy eHealth Applications in Cloud Computing Environment" Applied Sciences 11, no. 10: 4350. <https://doi.org/10.3390/app11104350>
7. Da Silva, M.A.P.; Times, V.C.; De Araujo, A.M.C.; Da Silva, P.C. A Microservice-Based Approach for Increasing Software Reusability in Health Applications. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.

8. Leal Sobral, Victor Ariel, Jacob Nelson, Loza Asmare, Abdullah Mahmood, Glen Mitchell, Kwadwo Tenkorang, Conor Todd, Bradford Campbell, and Jonathan L. Goodall. 2023. "A Cloud-Based Data Storage and Visualization Tool for Smart City IoT: Flood Warning as an Example Application" *Smart Cities* 6, no. 3: 1416-1434. <https://doi.org/10.3390/smartcities6030068>
9. Barthelemy, J.; Amirghasemi, M.; Arshad, B.; Fay, C.; Forehead, H.; Hutchison, N.; Iqbal, U.; Li, Y.; Qian, Y.; Perez, P. Problem-Driven and Technology-Enabled Solutions for Safer Communities: The case of stormwater management in the Illawarra-Shoalhaven region (NSW, Australia). In *Handbook of Smart Cities*; Springer: Berlin, Germany, 2020; pp. 1–28.
10. Shanmuga Sundaram, J.P.; Du, W.; Zhao, Z. A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues. *IEEE Commun. Surv. Tutor.* 2020, 22, 371–388.
11. Luft, Angela, Nils Luft, and Kristian Armtz. 2023. "A Basic Description Logic for Service-Oriented Architecture in Factory Planning and Operational Control in the Age of Industry 4.0" *Applied Sciences* 13, no. 13: 7610. <https://doi.org/10.3390/app13137610>
12. Perri, Damiano, Marco Simonetti, and Osvaldo Gervasi. 2022. "Deploying Efficiently Modern Applications on Cloud" *Electronics* 11, no. 3: 450. <https://doi.org/10.3390/electronics11030450>
13. Betarte, G.; Giménez, E.; Martínez, R.; Álvaro, P. Machine learning-assisted virtual patching of web applications. arXiv 2018, arXiv:cs.CR/1803.05529.
14. Schnieder, Maren, and Sheryl Williams. 2023. "Educational Mobile Apps for Programming in Python: Review and Analysis" *Education Sciences* 13, no. 1: 66. <https://doi.org/10.3390/educsci13010066>
15. Guedj, Benjamin, and Bhargav Srinivasa Desikan. 2020. "Kernel-Based Ensemble Learning in Python" *Information* 11, no. 2: 63. <https://doi.org/10.3390/info11020063>