

УДК 004.89

DOI <https://doi.org/10.32782/IT/2024-1-14>

Віра ШЕНДРИК

кандидат технічних наук, доцент, завідувач кафедри інформаційних технологій, Сумський державний університет, вул. Харківська, 116, м. Суми, Україна, 40007

ORCID: 0000-0001-8325-3115

Scopus Author ID: 54421163800

Юлія ПАРФЕНЕНКО

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій, Сумський державний університет, вул. Харківська, 116, м. Суми, Україна, 40007

ORCID: 0000-0003-4377-5132

Scopus Author ID: 54420952900

Сергій ТИМЧУК

доктор технічних наук, доцент, професор кафедри інформаційних технологій, Сумський державний університет, вул. Харківська, 116, м. Суми, Україна, 40007

ORCID: 0000-0002-8600-4234

Scopus Author ID: 55890931200

Ольга БОЙКО

кандидат технічних наук, доцент, старший викладач кафедри інформаційних технологій, Сумський державний університет, вул. Харківська, 116, м. Суми, Україна, 40007

ORCID: 0000-0001-8557-2267

Scopus Author ID: 54420952900

Олександр ГОРБАТЕНКО

магістрант факультету електроніки та інформаційних технологій, Сумський державний університет, вул. Харківська, 116, м. Суми, Україна, 40007

ORCID: 0009-0001-2042-3831

Бібліографічний опис статті: Шендрик, В., Парфененко, Ю., Тимчук, С., Бойко, О., Горбатенко, О. (2024). Мікросервіс хмарного зберігання та обміну даними моделей прогнозування системи підтримки прийняття рішень при управлінні гібридною енергомережею. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 1, 117–127, doi: <https://doi.org/10.32782/IT/2024-1-14>

**МІКРОСЕРВІС ХМАРНОГО ЗБЕРІГАННЯ ТА ОБМІНУ ДАНИМИ МОДЕЛЕЙ
ПРОГНОЗУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ
ПРИ УПРАВЛІННІ ГІБРИДНОЮ ЕНЕРГОМЕРЕЖЕЮ**

Метою роботи є розроблення мікросервісу для зберігання моделей прогнозування, які використовуються в системі підтримки прийняття рішень при управлінні гібридними енергетичними мережами, та їх метаданих. Це сприятиме упорядкуванню моделей прогнозування при їх зберіганні та підвищенню ефективності управління моделями прогнозування у хмарному сховищі.

Методологія. У даній роботі використано методологію системного аналізу, проектування інформаційних систем, організації зберігання даними та файлами у хмарних сховищах. Для розроблення та тестування мікросервісу використано метод прототипування, а саме програмної реалізації прототипу мікросервісу на основі RESTful API. Розгортання мікросервісу здійснювалося з використанням контейнеризації через Docker.

Наукова новизна роботи полягає у розробленні нового архітектурного рішення щодо зберігання та управління моделями прогнозування, розміщеними у хмарному сховищі S3. Спроектовано архітектуру підсистеми зберігання та обміну даними у вигляді мікросервісу з використанням архітектурного шаблону RESTful.

Висновки. У результаті проведеного дослідження було досягнуто наступних результатів. Розроблено об'єктне сховище моделей прогнозування S3, що відповідає архітектурі файлових сховищ хмарних

провайдерів. Розроблено мікросервіс, який реалізує API інтерфейс обробки запитів до сховища моделей прогнозування та забезпечує управління моделями прогнозування, збереженими у сховищі. Розроблений мікросервіс буде інтегрований із системою підтримки прийняття рішень при управлінні енергомережами і може використовуватися як локально, так і розгортатися на платформах провідних провайдерів хмарних сервісів. Проведено тестування мікросервісу з використанням нейромережових прогнозних моделей електроспоживання різних типів, яке підтвердило його працездатність.

Ключові слова: хмарні технології, мікросервіс, нейронна мережа, моделі прогнозування, об'єктне сховище, інтеграція даних, підтримка прийняття рішень.

Vira SHENDRYK

PhD, Associate Professor, Head at Information Technology Department, Sumy State University, 116, Kharkivska Str., Sumy, Ukraine, 40007, v.shendryk@cs.sumdu.edu.ua

ORCID: 0000-0001-8325-3115

Scopus Author ID: 54421163800

Yuliia PARFENENKO

PhD, Associate Professor, Associate Professor at Information Technology Department, Sumy State University, 116, Kharkivska Str., Sumy, Ukraine, 40007, yuliya_p@cs.sumdu.edu.ua

ORCID: 0000-0003-4377-5132

Scopus Author ID: 54420952900

Sergii TYMCHUK

D.Sc, Associate Professor, Professor at Information Technology Department, Sumy State University, 116, Kharkivska Str., Sumy, Ukraine, 40007, s.tymchuk@itp.sumdu.edu.ua

ORCID: https://orcid.org/0000-0002-8600-4234

Scopus Author ID: 55890931200

Oliha BOIKO

PhD, Associate Professor, Senior lecturer at Information Technology Department, Sumy State University, 116, Kharkivska Str., Sumy, Ukraine, 40007, yuliya_p@cs.sumdu.edu.ua

ORCID: 0000-0001-8557-2267

Scopus Author ID: 54420952900

Olexander HORBATENKO

Master's student of the Electronics and Information Technologies Faculty, Sumy State University, 116, Kharkivska Str., Sumy, Ukraine, 40007, shuric91@gmail.com

ORCID: 0009-0001-2042-3831

To cite this article: Shendryk, V., Parfenenko, Yu., Tymchuk, S., Boiko, O., Horbatenko, S. (2024). Mikroservis khmarnoho zberihannia ta obminu danymy modelei prohnozuvannia systemy pidtrymky pryiniattia rishen pry upravlinni hibrydnoi enerhomerezheiu [Microservice of forecasting models cloud storage and data exchange at the decision support system for hybrid power network management]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 1, 117–127, doi: <https://doi.org/10.32782/IT/2024-1-14>

MICROSERVICE OF FORECASTING MODELS CLOUD STORAGE AND DATA EXCHANGE AT THE DECISION SUPPORT SYSTEM FOR HYBRID POWER NETWORK MANAGEMENT

The purpose of the study is to develop a microservice for storing forecasting models used in the decision support system for energy management of hybrid power grids and their metadata. This will help organize forecasting models when they are stored and increase the efficiency of managing forecasting models in cloud storage.

Methodology. This work uses the methodology of system analysis, design of information systems, organization of data and file storage in cloud storage. The prototyping method was used for the development and testing of the microservice, namely the software implementation of the microservice prototype based on the RESTful API. Microservice deployment was done using containerization through Docker.

The scientific novelty of the study consists in the development of a new architectural solution for the storage and management of prediction models placed in the S3 cloud storage. The architecture of the storage and data exchange subsystem was designed in the form of a microservice using the RESTful architectural template.

Conclusions. *The following results were achieved. The S3 object storage of forecasting models have been developed, which corresponds to the file storage architecture of cloud providers. A microservice has been developed that implements the API interface for processing requests to the storage of prediction models and provides management of the forecasting models stored in the storage. The developed microservice will be integrated with the decision support system for power grid management and can be used both locally and deployed on the platforms of leading cloud service providers. The microservice was tested using neural network forecasting models of various types of electricity consumption, which confirmed its functionality.*

Key words: *cloud technologies, microservice, neural network, prediction models, object storage, data integration, decision support.*

Актуальність проблеми. Енергетична безпека, стійкість та підвищення енергоефективності є ключовими факторами досягнення сталого розвитку в усьому світі. Нагальною потребою є скорочення використання викопного палива і збільшення частки відновлюваних джерел енергії (Ostergaard, P., Noorollahi, Y. & Kalogirou, S., 2022). Для України сьогодні вкрай важливою є енергоефективно орієнтована інноваційна модернізація енергетики з тактичним і стратегічним оновленням економіки (Кириленко, О. В. та ін., 2023). Під тактичним оновленням розуміємо використання вже розроблених і апробованих на практиці технологій, які можуть бути впроваджені за короткий термін, а під стратегічним – проведення відповідних фундаментальних досліджень і розроблення за їх результатами нових технологій, зокрема створення інтелектуальних мереж електро-, тепло- і водозабезпечення у відповідності до концепції «розумних» міст (Obukhov, S. et al., 2019). Ефективне управління енергоспоживанням потребує використання системи енергетичного менеджменту, що складається з комплексу програмних і апаратних засобів (Shendryk, V. et al., 2021, Shendryk, S. et al., 2021). В системах підтримки прийняття рішень при управлінні енергозабезпеченням використовуються як дані моніторингу функціонування енергосистеми, так і прогнозні дані виробництва та споживання енергоресурсів. Процеси прогнозування та оптимізації енерговитрат супроводжуються обробкою великих масивів даних, чим більше вибірка даних, тим точніша модель прогнозування, а отже правильніші прийняті управлінські рішення (Shendryk, V. et al., 2022). Для прогнозування генерації електроенергії, у тому числі від альтернативних джерел, а також енергоспоживання використовуються попередньо розроблені моделі на різні часові інтервали та з різними параметрами. Отже в системах підтримки прийняття рішень використовуються ансамблі моделей прогнозування. Необхідність підтримки керування файловими даними та метаданими в середовищах великих даних зумовлена необхідністю їх використання у системах прийняття рішень, коли маємо багато

неструктурованих моделей прогнозування разом з їх конфігураціями та іншими параметрами (Smith, K.P. et al., 2014).

В системах підтримки прийняття рішень при управлінні енергетичною інфраструктурою використовуються дані з різних джерел, у тому числі й значення прогнозу споживання та генерації енергії для різних об'єктів та на різні часові інтервали. Це вимагає використання набору моделей прогнозування, що зумовлює потребу в упорядкуванні їх зберігання та управління моделями. Також для запуску моделей необхідно зберігати службову інформацію у вигляді метаданих, як наприклад тип моделі чи довжина інтервалу прогнозування. Все більш широкого використання набуває організація зберігання файлів у вигляді об'єктного сховища. Цей підхід може бути використаний і для зберігання файлів нейронних мереж прогнозування енергоспоживання.

Актуальною є задача організації зберігання моделей прогнозування таким чином, щоб забезпечити їх оновлення, ефективно використання даних моделей та інших системних даних, що потребує розроблення сервісу управління сховищем даних.

Аналіз останніх досліджень і публікацій.

Протягом останнього десятиліття необхідність боротьби з надмірним обсягом даних сприяла появі нових систем зберігання великих даних, які відійшли від традиційних моделей реляційних баз даних (Joseph S. Valacich et al., 2022). Ці підходи зазвичай жертвують такими властивостями, як узгодженість даних, щоб підтримувати швидкі відповіді на запит із збільшенням обсягів даних (Saiqa Aleem et al., 2017). Сховища даних та файлові сховища використовуються в системах підтримки прийняття рішень подібно до традиційних систем керування реляційними базами даних, забезпечуючи збереження та обробку неструктурованих та напівструктурованих даних, а також файлів.

З появою технології Інтернету речей (IoT) зріс попит на використання розподілених комп'ютерних ресурсів та використання безпечних та стабільних мережевих послуг (Fei Hu, 2016). З різними видами обчислювальних

послуг і розвитком інформаційних технологій хмарні технології, що включають програми або послуги, що надаються через Інтернет, апаратні компоненти, системи зберігання та системне програмне забезпечення, все частіше використовуються для віддаленого зберігання та управління даними (Antoni Olivé, 2017, Roel J. Wieringa, 2014, Raul Gracia-Tinedo et al., 2013).

Хмарне сховище забезпечує гнучкий доступ із багатьох місць і швидко й просто масштабування ємності, а також нижчі ціни на сховище та кращу підтримку порівняно з власним корпоративним сховищем (Greg Schulz, 2011).

Існують різні способи зберігання даних у хмарі – блокове, об'єктне та файлове сховище. Об'єктне сховище зберігає та керує всіма даними в неструктурованому форматі та в одиницях, які називаються об'єктами. Блокове сховище приймає будь-які дані, наприклад файл або запис бази даних, і розділяє їх на блоки однакового розміру. Потім воно зберігає блок даних у фізичному сховищі, оптимізованому для швидкого доступу та пошуку. Файлове сховище – метод зберігання даних, який надає серверам і програмам доступ до даних через спільні файлові системи.

Блокове сховище забезпечує високу швидкість обробки даних, низьку затримку та високу продуктивність. Будь-яка хмарна служба, яка потребує швидкого доступу до даних, працює з блоковим сховищем. Наприклад, аналітика в режимі реального часу, високопродуктивні обчислення та системи з великою кількістю швидких транзакцій – усі вони виграють від блокового зберігання. Хмарне сховище файлів найкраще, коли користувачам потрібен одночасний доступ до спільної системи файлів. Крім того, контроль доступу на рівні файлів дозволяє налаштувати дозволи та списки контролю доступу (ACL) для підвищення безпеки. Наприклад, робочі середовища для спільної роботи, які потребують спільного використання файлів між віддаленими командами, використовують сховище файлів.

Проаналізовано рекомендації щодо вибору хмарного сховища від одного з провайдерів хмарних сервісів Google Cloud Service (cloud.google.com, 2024). Основним критерієм при виборі типу сховища за вказаною схемою є протокол доступу до даних. Для організації зберігання та обміну даними моделей прогнозування необхідним є використання протоколу доступу RESTful, тому об'єктне сховище є переважним (Carullo, Giulian et al., 2017).

Об'єктне сховище окрім неструктурованих файлових даних, може також зберігати

метадані, у випадку моделей прогнозування їх конфігурацію та список залежностей. Таке сховище ще є більш економічно вигідне, при цьому маючи високу можливість масштабування, а також враховуючи рекомендації від Google Cloud Service, найоптимальнішим типом хмарного сховища для вирішення поставлених задач (Factor, Michael et al., 2005, Pawar, S. et al. 2020).

Окрім зберігання моделей прогнозування, необхідним є управління сховищем для організації упорядкування, оновлення моделей, видалення тих, які не використовуються. Аналіз підходів до розробки програмного забезпечення показав, що для незалежного розгортання та можливості інтеграції в систему підтримки прийняття рішень для вирішення цієї задачі необхідним є використання мікросервісного підходу (Lagucea, Xabier, 2018).

Мета дослідження – розробка мікросервісу зберігання моделей прогнозування та їх конфігураційних даних, яка буде використовуватися в системі підтримки прийняття рішень при управлінні гібридною енергомережею. Необхідно забезпечити організацію зберігання прогнозних моделей, які представляють собою бінарні файли, так і метаданих, такі як конфігурація моделей, а також залежностей, використовуваних в процесі прогнозування.

Виклад основного матеріалу дослідження. Архітектурним рішенням для вирішення поставленої задачі є розроблення мікросервісу в поєднанні з S3-подібним об'єктним сховищем даних, яке можна розгорнути як локально за допомогою контейнеризації, так і в якості service-mesh. Використання технології мікросервісів дозволить скоротити час на розробку рішення, дозволить уніфікувати інтерфейс взаємодії з іншими системами та підсистемами. Також такий підхід дає змогу розробити платформо-незалежний додаток, що дозволить використовувати стек технологій, відмінних від використаних в системі підтримки прийняття рішень.

Проаналізувавши сучасні підходи до розроблення та проектування інформаційних систем, а також інтеграції даних між їх складовими, було встановлено такі функціональні вимоги до підсистеми зберігання та обміну моделей прогнозування, реалізованої як мікросервіс:

- підсистема повинна зберігати файли моделей разом з їх метаданими та надавати відповідні файли та дані за запитом;
- підсистема повинна надавати швидкий доступ до найбільш використовуваних моделей, що можна забезпечити за допомогою кешування;



Рис. 1. Архітектура мікросервісу

- сховище даних повинно бути об'єктним, а також S3-сумісним для можливості його розгортання у хмарній інфраструктурі;
- підсистема повинна мати загальну доступність у будь-який момент часу, під високим навантаженням, з низькою затримкою, високою стійкістю та гнучкістю, повинна підтримувати RESTful доступ.

Для моделювання процесів взаємодії підсистеми зберігання та обміну даними, реалізованої як мікросервіс, із сховищем моделей та системою підтримки прийняття рішень, розроблено архітектурну модель, яку представлено за допомогою нотації C4.

Система підтримки прийняття рішень представляє собою окремий вебсервіс, який запускає з певною періодичністю аналіз вхідних даних, зібраних з електромереж різних типів та різних конфігурацій. Цей сервіс потребує доступу до попередньо натренованих моделей прогнозування, а також збереження таких моделей та інших даних.

Підсистема зберігання та обміну даними – мікросервіс, який надає доступ до моделей прогнозування з використанням REST API запитів.

Діаграму варіантів використання розробленого мікросервісу показано на рис. 2. Виділено таких акторів:

- Системний адміністратор – може завантажувати прогнозні моделі в S3 сховище, керувати метаданими, налаштуванням доступу, створенням нових Buckets тощо;
- Модуль активації – модуль, який здійснює запити на отримання метаданих та файлових даних моделі прогнозування, а також завантажувати чи оновлювати моделі;
- MinIO S3 сервер – провайдер S3 об'єктного сховища;

- Cache Redis – сховище, яке дає швидкий доступ до найбільш часто запитуваних моделей, організовуючи кешування об'єктів.

MinIO – сервіс-провайдер доступу до об'єктного сховища на базі протоколу S3. Даний сервіс також розгортається з використанням середовища Docker. MinIO дозволяє розгортання декількох мікросервісів, об'єднаних у кластер, але в даній роботі вибрано варіант розгортання даного продукту як окремого сервісу в одному Docker-контейнері.

Models Repository – S3-подібне об'єктне сховище даних. В роботі було використано Docker volumes в якості фізичного дискового простору, але є можливість підключення зовнішніх дисків та пристроїв, а також використання сховища даних від хмарних провайдерів.

Cache використовується для покращення швидкодії та доступу. Для моделей, які часто викликаються, використовується механізм in-memory кешування за допомогою інструментів Redis. Кожна модель прогнозування після отримання файлу моделі з S3 сховища поміщується у сховище Redis, де зберігається протягом 30 хвилин, та у випадку подальшого запиту на отримання даної моделі, буде зроблена перевірка співпадіння хеш суми файлу моделі в кеші та хеш суми об'єкту в S3 сховища, яка відповідає даному файлу. Якщо їх хеш суми співпадуть, буде надано дані з кешу Redis, і не буде робитися запит до MinIO. При повторному запиті час життя моделі в сховищі Redis автоматично подовжується на 30 хвилин. Таким чином реалізується кешування моделі прогнозування у сховищі Redis, що дозволяє забезпечити швидкий доступ до моделей прогнозування, які часто використовуються.

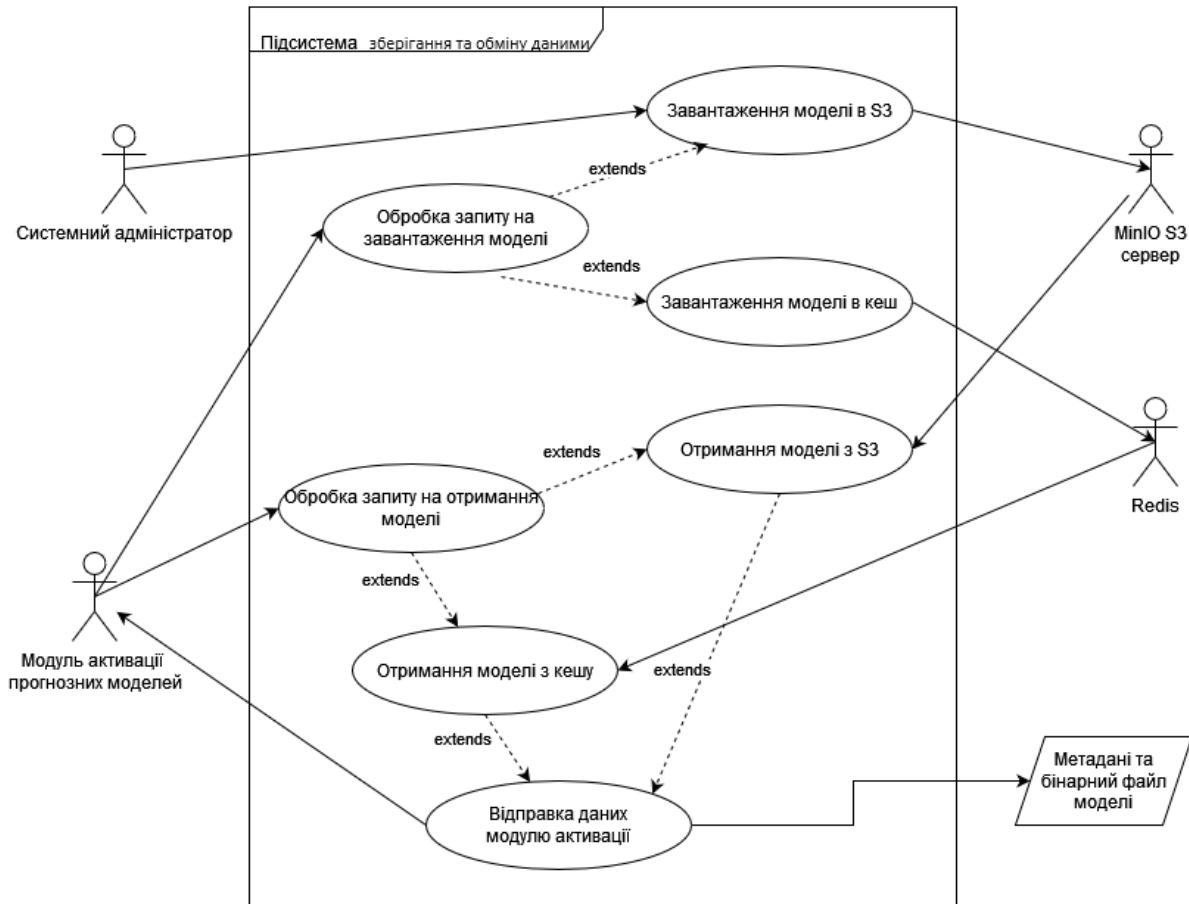


Рис. 2. Діаграма варіантів використання мікросервісу

Процес отримання даних про моделі прогнозування виконується наступним чином. Зовнішня система або користувач надсилає запит на отримання моделі з параметрами `user_id`, `data_source_id`, `model_type`, `model_name`. Система робить запит до Redis і перевіряє, чи запитувана модель знаходиться в кеші. Якщо так, то відбувається запит до MinIO на отримання метаданих. Якщо метаданих за запитом не існує, то дані видаляються з кешу і користувачу повертається HTTP 404 Not Found. Якщо метадані існують, вони повертаються користувачу разом з тимчасовим посиланням на файл моделі. Якщо кеш не містить даних про модель, система робить запит MinIO і повертає їх користувачу разом з тимчасовим посиланням на файл моделі, а в разі їх відсутності повертає HTTP 404 Not Found. Зовнішня система (модуль активації) робить запит на отримання файлу моделі за тимчасовим посиланням.

Процес завантаження моделей прогнозування та їх метаданих даних складається з наступної послідовності кроків. Зовнішня система (або користувач) надсилає запит на завантаження моделі з параметрами `user_id`, `data_source_id`, `model_type`, `model_name`.

Система надсилає запит до MinIO на створення метаданих, і якщо вони були успішно поміщені в S3 сховище, то повертає їх користувачу разом з тимчасовим посиланням на завантаження файлу моделі. Зовнішня система (користувач) робить запит на завантаження файлу моделі за тимчасовим посиланням. Система завантажує файл в MinIO і зв'язує її з створеними метаданими, розміщує файл та метадані моделі прогнозування в кеш.

Розробка мікросервісу проводилася в середовищі IntelliJ Idea з використанням Spring Boot версії 2.7. Середовищем для розгортання мікросервісу обрано Docker. В якості провайдера S3 об'єктного сховища обрано сервіс MinIO, який доступний у вигляді Docker Image в публічному репозиторії. Інструменти Redis, який застосовується для кешування файлових даних моделей, а також nginx, який використовується як проксі-сервер для Minio, також є в публічному доступі і готові для розгортання у середовищі Docker. Для запуску перелічених сервісів використано docker compose. Для зберігання моделей прогнозування використано S3 Bucket. Результат запуску створених контейнерів за допомогою docker compose показано на рис. 3.

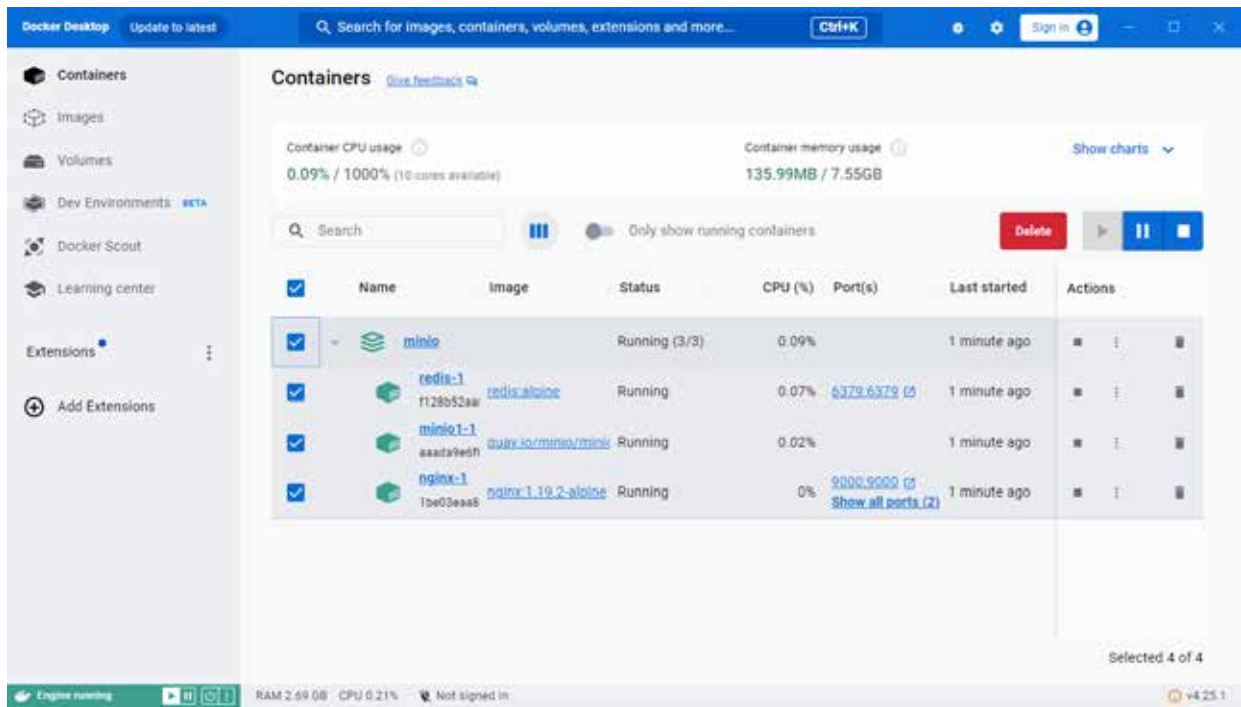


Рис. 3. Docker контейнери після запуску

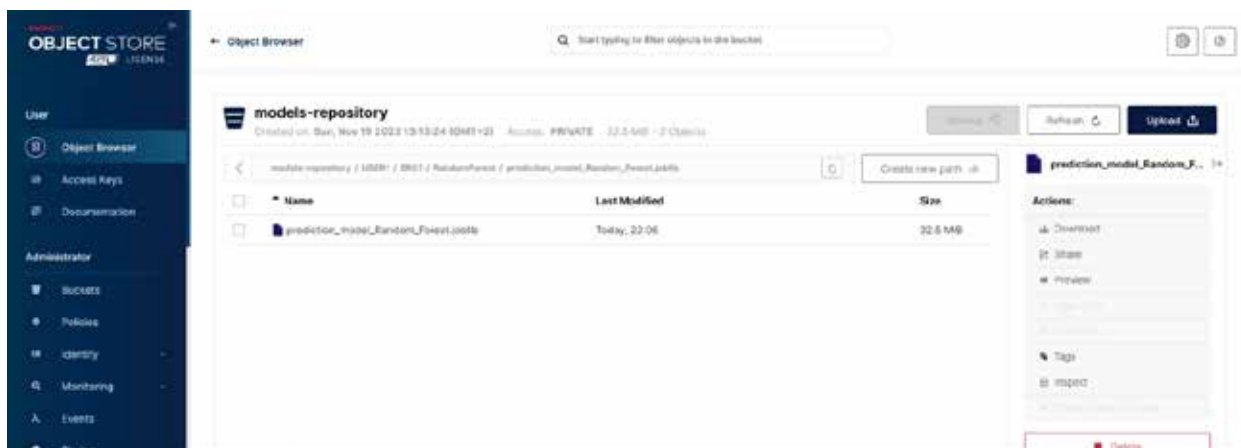


Рис. 4. Збережений файл моделі у об'єктному сховищі

Після додавання Bucket його можна переглянути у вікні адміністрування MinIO. Після виконання вищеописаних кроків об'єктне сховище S3 налаштоване і готове для використання. Приклад доступу до збереженої моделі прогнозування засобами MinIO показано на рис. 4.

Мікросервіс, який реалізує API інтерфейс обробки запитів до сховища моделей, реалізовано мовою Java з використанням об'єктно-орієнтованого підходу. Нижче приведено опис основних класів та їх призначення.

`ModelsStorageServiceApplication` – головний клас, що є початковою точкою запуску додатку.

`ObjectsStorageController` – клас, який реалізує REST-контролер, що дозволяє обробляти

запити на створення та отримання метаданих моделей.

`FilesController` – клас, який реалізує REST-контролер, що дозволяє обробляти запити на завантаження та скачування бінарних файлів моделей.

`ModelsStorageService` – інтерфейс, який описує основні функції бізнес-логіки.

`MinIoModelsStorageServiceImpl` – клас, який реалізує основну бізнес-логіку сервісу. У цьому класі здійснюється підключення до сервісу MinIO, перетворення метаданих у відповідності до стандарту S3, збереження та зчитування метаданих та файлів моделей з використанням MinIO клієнта, кешування.

```

1  Body  Cookies  Headers (9)  Test Results
2  Pretty  Raw  Preview  Visualize  JSON  [ ]
3  1
4  2  "modelName": "prediction_model_LSTM.joblib",
5  3  "modelFullName": "USER1/SRC1/LSTM/prediction_model_LSTM.joblib",
6  4  "configuration": {
7  5      "module": "keras",
8  6      "class_name": "Sequential",
9  7      "config": {
10 8          "name": "sequential_2",
11 9          "layers": [
1210              {
1311                  "module": "keras.layers",
1412                  "class_name": "InputLayer",
1513                  "config": {
1614                      "batch_input_shape": [
1715                          null,
1816                          1.0,
1917                          9.0
2018                      ],
2119                      "dtype": "float32",
2220                      "sparse": false,
2321                      "ragged": false,
2422                      "name": "lstm_2_input"
2523                  }
2624              },
2725              {
2826                  "module": "keras.layers",
2927                  "class_name": "LSTM",
3028                  "config": {
3129                      "name": "lstm_2",

```

Рис. 5. Результат отримання метаданих моделі прогнозування

Таблиця 1

Результати часу завантаження бінарних файлів

	№ запиту	Модель прогнозування, 162.9 КБ	Модель прогнозування, 32 МБ
З увімкненим кешуванням	1	126	688
	2 (відразу за 1-им)	9	440
	3 (після 1 хв. за 1-им)	10	455
Без увімкненого кешування	1	12	698
	2 (відразу за 1-им)	13	462
	3 (після 1 хв. за 1-им)	10	455

Для того, щоб запустити реалізований мікросервіс із середовища розробки IntelliJ Idea, необхідно обрати головний клас додатку ModelsStorageServiceApplication та виконати команду «Run».

Щоб додати модель в об'єктне сховище S3, потрібно спочатку відправити запит на створення моделі. Якщо в моделі немає метаданих, вони створюються в у сховищі S3, якщо існують, то будуть оновлені. Після успішного створення чи оновлення метаданих моделі повертається S3 ідентифікатор моделі.

Мікросервіс виконує запити на завантаження бінарних файлів моделей прогнозування. Після успішного завантаження файлу

надсилається eTag ідентифікатор сховища S3. Після успішно виконаних запитів модель як об'єкт буде доступна в S3 сховищі, її можна переглянути в адміністративній консолі MinIO. Також доступні для перегляду метадані моделі, в яких описана службова інформація, що вказує на шлях до файлу моделі, тип моделі прогнозування, її конфігураційні параметри тощо.

Було виконане тестування розробленого мікросервісу на декількох нейромережових моделях прогнозування з LSTM архітектурою. Досліджено результати часу завантаження бінарних файлів моделей прогнозування в режимі з кешуванням та без кешування, які наведено в табл. 1.

Встановлено, що перші запити на отримання моделей прогнозування з увімкненим кешуванням є дещо повільнішим за запити з вимкненим кешуванням. Це пов'язано з тим, що системі потрібен час на відправку даних в Redis. Подальші запити є швидшими, що забезпечується використанням кешованих даних.

Висновки і перспективи подальших досліджень. Проведено аналіз існуючих рішень та технологій обміну та зберігання даних у хмарному сховищі необхідних для розробки підсистеми зберігання та обміну даними системи підтримки прийняття рішень при управлінні гібридною енергомережею. Також було формалізовано основні функціональні вимоги до цієї підсистеми. В результаті порівняльного аналізу основних типів хмарних сховищ (об'єктне, блочне та файлове) було встановлено, що об'єктне сховище найбільше підходить для вирішення поставлених задач, оскільки воно забезпечує високу масштабованість, доступність та економічність. Крім того, об'єктне сховище дозволяє зберігати не тільки файлові дані моделей прогнозування, а також і набір метаданих, таких як конфігурація, залежності, тощо.

В ході роботи були розглянуті сучасні архітектурні підходи, фреймворки, технології та інструменти розробки. Було спроектовано архітектуру підсистеми з використанням RESTful архітектурного шаблону, Spring Boot мікросервісів та протоколу доступу до даних об'єктного сховища S3. Практичне значення роботи полягає у тому, що розроблена підсистема зберігання та обміну моделей побудована за мікросервісною архітектурою з використанням S3 протоколу для обміну даними з об'єктним сховищем, стане частиною системи підтримки прийняття рішень управління гібридними електромережами яку можна буде розгорнути в хмарному сервісі з мінімальними доопрацюваннями. Розробка має практичне значення для використання у системах підтримки прийняття рішень, оскільки сприяє підвищенню ефективності управління моделями прогнозування.

Розроблений мікросервіс може бути використаний в системі підтримки прийняття рішень при управлінні енергомережами, де реалізоване як локальне зберігання моделей прогнозування, так із використанням провідних хмарних сервісів.

ЛІТЕРАТУРА:

1. Ostergaard Poul, Duic Neven, Noorollahi Younes, Kalogirou Soteris. Renewable energy for sustainable development. *Renewable Energy*. 2022. Vol. 199. P. 1145–1152. <https://doi.org/10.1016/j.renene.2022.09.065>
2. Кириленко, О. В., Снежкін, Ю. Ф., Басок, Б. І., Базєєв, Є. Т. Енергетика, наука та інженерія: сучасний стан і виклики розвитку. *Вісник НАН України*. 2023. № 4, с. 3–20. <https://doi.org/10.15407/visn2023.04.003>
3. Obukhov S., Ibrahim A., Tolba M.A., M.El-Rifaie A. Power balance management of an autonomous hybrid energy system based on the dual-energy storage. *Energies*. 2019. Vol.12. <https://doi.org/10.3390/en12244690>.
4. Shendryk V., Boiko O., Parfenenko Y., Shendryk S., Tymchuk S. Decision Making for Energy Management in Smart Grid. *Research Anthology on Clean Energy Management and Solutions*. 2021. P. 1742–1776. <https://doi.org/10.4018/978-1-7998-9152-9.ch077>
5. Shendryk S., Shendryk V., Parfenenko Y., Drozdenko O., Tymchuk S. Decision Support System for Efficient Energy Management of MicroGrid with Renewable Energy Sources. *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*. 2021. P. 225–230. <https://doi.org/10.1109/IDAACS53288.2021.9660966>.
6. Shendryk V., Parfenenko Y., Tymchuk S., Kholiavka Y., Bielka Y. Modeling techniques of electricity consumption forecasting. *AIP Conference Proceedings*. 2022. Vol. 2570. <https://doi.org/10.1063/5.0100123>
7. Smith K.P., Seligman L.J., Rosenthal A., Kurcz C., Greer M., Macheret C., Sexton M., Eckstein A. Big metadata: The need for principled metadata management in big data ecosystems. *Proceedings of the Third Workshop on Data analytics in the Cloud, DanaC*. 2014, Snowbird, Utah, USA, pp. 13:1–13:4. <https://doi.org/10.1145/2627770.2627776>
8. Joseph S. Valacich, Christoph Schneider, Matthew Hashim. *Information systems today: managing in the digital world*. Pearson Education, 2022. 568 p.
9. Saiqa Aleem, Rabia Batool, Faheem Ahmed, Asad Khatak, and Raja Muhammad Ubaid Ullah. Architecture guidelines for saas development process. *Proceedings of the 2017 International Conference on Cloud and Big Data Computing, ICCBDC 2017*. 2017. P. 94–99.
10. Fei Hu. *Big Data: Storage, Sharing, and Security*. CRC Press, 2016. 430 p.
11. Antoni Olivé. *Conceptual modeling of information systems*. Springer Berlin Heidelberg, 2017. 455 p.
12. Roel J. Wieringa. *Design science methodology: For information systems and software engineering*. Springer Berlin Heidelberg, 2014. 332 p.

13. Raul Gracia-Tinedo, Marc Sanchez Artigas, Adrian Moreno-Martinez, Cristian Cotes, and Pedro Garcia Lopez. Actively measuring personal cloud storage. *In 2013 IEEE Sixth International Conference on Cloud Computing*, 2013. P.301–308.
14. Greg Schulz. *Cloud and Virtual Data Storage Networking*. Auerbach Publications, 2011. 400 p.
15. Design an optimal storage strategy for your cloud workload. (n.d.). *cloud.google.com*. URL:<https://cloud.google.com/architecture/storage-advisor>
16. Carullo, Giuliano & Di Mauro, Mario & Galderisi, Michele & Longo, Maurizio & Postiglione, Fabio & Tambasco, Marco. Object Storage in Cloud Computing Environments: An Availability Analysis. 2017. P. 178–190. https://10.1007/978-3-319-57186-7_15.
17. Factor, Michael & Meth, Kalman & Naor, Dalit & Rodeh, Ohad & Satran, Julian. Object storage: The future building block for storage systems. *IEEE Xplore*, 2005. P.119. <https://doi.org/10.1109/LGDI.2005.1612479>
18. Pawar S., Bhusari C., Gopnarayan A. A Study of Google Cloud Services. *International Journal of Engineering Applied Sciences and Technology*. 2020. Vol. 5, Issue 7, No. 2455-2143, P. 227–231. <https://doi.org/10.33564/IJEAST.2020.V05I07.034>
19. Larrucea, Xabier & Santamaria, Izaskun & Colomo-Palacios, Ricardo & Ebert, Christof. *Microservices*. *IEEE Software*, 2018. Vol. 35. P. 96–100. <https://doi.org/10.1109/MS.2018.2141030>

REFERENCES:

1. Ostergaard, Poul, Duic, Neven, Noorollahi, Younes & Kalogirou, Soteris. (2022). Renewable energy for sustainable development. *Renewable Energy*, Vol. 199, p. 1145–1152. <https://doi.org/10.1016/j.renene.2022.09.065>
2. Kyrylenko, O.V., Snezhkin, Y.F., Basok, B.I. & Bazyeev, Y.T. (2023). Ukraine's energy: probable scenarios of recovery and development. *Visn. Nac. Akad. Nauk Ukr.*, (4), 3–20. <https://doi.org/10.15407/visn2023.04.003> [in Ukrainian]
3. Obukhov, S., Ibrahim, A., Tolba, M.A. & M.El-Rifaie, A. (2019). Power balance management of an autonomous hybrid energy system based on the dual-energy storage. *Energies*, 2019, v.12. <https://doi.org/10.3390/en12244690>.
4. Shendryk, V., Boiko, O., Parfenenko, Y., Shendryk, S. & Tymchuk, S. (2021). Decision Making for Energy Management in Smart Grid. *Research Anthology on Clean Energy Management and Solutions*, p. 1742–1776. <https://doi.org/10.4018/978-1-7998-9152-9.ch077>
5. Shendryk, S., Shendryk, V., Parfenenko, Y., Drozdenko, O. & Tymchuk, S. (2021). Decision Support System for Efficient Energy Management of MicroGrid with Renewable Energy Sources. *Proceedings of the 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*. P. 225–230. <https://doi.org/10.1109/IDAACS53288.2021.9660966>.
6. Shendryk, V., Parfenenko, Y., Tymchuk, S., Kholiavka, Y. & Bielka, Y. (2022). Modeling techniques of electricity consumption forecasting. *AIP Conference Proceedings*. 2570. <https://doi.org/10.1063/5.0100123>
7. Smith, K.P., Seligman, L.J., Rosenthal, A., Kurcz, C., Greer, M., Macheret, C., Sexton, M. & Eckstein, A. (2014). Big metadata: The need for principled metadata management in big data ecosystems. *Proceedings of the Third Workshop on Data analytics in the Cloud, DanaC 2014, Snowbird, Utah, USA*, pp. 13:1–13:4. <https://doi.org/10.1145/2627770.2627776>
8. Joseph, S. Valacich, Christoph Schneider, Matthew Hashim. (2022). *Information systems today: managing in the digital world*. Pearson Education, 568 p.
9. Saiqa, Aleem, Rabia, Batool, Faheem, Ahmed, Asad, Khatak, and Raja Muhammad Ubaid Ullah. (2017). Architecture guidelines for saas development process. *Proceedings of the 2017 International Conference on Cloud and Big Data Computing, ICCBDC 2017*, p. 94–99.
10. Fei, Hu. (2016). *Big Data: Storage, Sharing, and Security*. CRC Press, 430 p.
11. Antoni Olivé. (2017). *Conceptual modeling of information systems*. Springer Berlin Heidelberg, 455 p.
12. Roel J. Wieringa. (2014). *Design science methodology: For information systems and software engineering*. Springer Berlin Heidelberg, 332 p.
13. Raul Gracia-Tinedo, Marc Sanchez Artigas, Adrian Moreno-Martinez, Cristian Cotes, and Pedro Garcia Lopez (2013). Actively measuring personal cloud storage. *In 2013 IEEE Sixth International Conference on Cloud Computing*, p.301–308
14. Schulz, Greg. (2011). *Cloud and Virtual Data Storage Networking*. Auerbach Publications, 400 p.
15. Design an optimal storage strategy for your cloud workload. (n.d.). *cloud.google.com*. Retrieved from <https://cloud.google.com/architecture/storage-advisor>

16. Carullo, Giuliano & Di Mauro, Mario & Galderisi, Michele & Longo, Maurizio & Postiglione, Fabio & Tambasco, Marco. (2017). Object Storage in Cloud Computing Environments: An Availability Analysis. 178–190. https://10.1007/978-3-319-57186-7_15.
17. Factor, Michael & Meth, Kalman & Naor, Dalit & Rodeh, Ohad & Satran, Julian. (2005). Object storage: The future building block for storage systems. *IEEE Xplore*, p.119 – <https://doi.org/10.1109/LGDI.2005.1612479>
18. Pawar, S., Bhusari, C. & Gopnarayan, A. (2020). A Study of Google Cloud Services. *International Journal of Engineering Applied Sciences and Technology*. Vol. 5, Issue 7, No. 2455–2143, P. 227–231. <https://doi.org/10.33564/IJEAST.2020.V05I07.034>
19. Larrucea, Xabier, Santamaria, Izaskun, Colomo-Palacios, Ricardo & Ebert, Christof. (2018). Microservices. *IEEE Software*. 35. 96–100. <https://doi.org/10.1109/MS.2018.2141030>