Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Метод виявлення аномалій у роботі системи обробки даних за допомогою моделей глибокого навчання
(тема)

Виконав:
здобувач 2 року навчання,
групи СПм-23-4
Олексій НІКОЛАЄВ
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: доц. Ірина ІЛЬЇНА
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп'ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
"____" _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ніколаєву Олексію Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Метод виявлення аномалій у роботі системи обробки даних за допомогою моделей глибокого навчання

затверджена наказом по університету від "21" квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи Набір даних

4. Перелік питань, що потрібно опрацювати у роботі

Особливості часових рядів у задачах виявлення аномалій

Архітектура та принцип роботи моделей CNN, LSTM, CNN+LSTM

Побудова моделі прогнозування часових рядів з використанням CNN+LSTM

Побудова базових моделей (наприклад, MA, LSTM) для порівняння

Порівняльна оцінка моделей

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 21 слайд

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Огляд методів виявлення аномалій в	22.04.25-29.04.25	
	системах обробки даних		
2	Вибір та обгрунтування методики дослідження	30.04.25-05.05.25	
3	Вибір інструментальних засобів	06.05.25-09.05.25	
4	Розробка моделей	10.05.25-20.05.25	
5	Проведення експериментів	21.05.25-02.06.25	
6	Оформлення матеріалів кваліфікаційної роботи	03.06.25-05.06.25	
7	Подання кваліфікаційної роботи керівникові та	06.06.25-09.06.25	
	її попередній захист		
8	Подання кваліфікаційної роботи на рецензу-	10.06.25-12.06.25	
	вання		

Дата видачі завдання " 21 " квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Ірина ІЛЬЇНА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 59 с., 13 рис., 6 табл., 1 дод., 9 джерел.

ГЛИБОКЕ НАВЧАННЯ, ВИЯВЛЕННЯ АНОМАЛІЙ, CNN, LSTM, ЧАСОВІ РЯДИ, ОБРОБКА ДАНИХ, ПРОГНОЗУВАННЯ, НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, СИСТЕМИ МОНІТОРИНГУ, MAE, RMSE, AUC, КЛАСИФІКАЦІЯ.

Метою кваліфікаційної роботи є розробка та дослідження методу виявлення аномалій у роботі системи обробки даних із використанням моделей глибокого навчання, зокрема гібридної архітектури CNN+LSTM.

У ході виконання кваліфікаційної роботи було здійснено аналіз структури системи обробки даних, підготовлено часові ряди на основі логів активності, виявлено сезонні та нестабільні характеристики даних. Реалізовано кілька моделей прогнозування та аномалійної класифікації, включаючи MA, LSTM та CNN+LSTM. Проведено експерименти з різною кількістю історичних вхідних днів для оптимізації точності. Розроблено механізм виявлення аномалій на основі прогнозної похибки та довірчого інтервалу. Отримані результати свідчать про високу ефективність гібридного підходу для задачі моніторингу й оцінки стану інформаційних систем у режимі, близькому до реального часу.

ABSTRACT

Master’s thesis: 59 pages, 13 figures, 6 tables, 1 appendices, 9 sources.

DEEP LEARNING, ANOMALY DETECTION, CNN, LSTM, TIME SERIES, DATA PROCESSING, FORECASTING, NEURAL NETWORKS, MACHINE LEARNING, MONITORING SYSTEMS, MAE, RMSE, AUC, CLASSIFICATION.

The major goal of this thesis is the development and study of a method for detecting anomalies in the operation of a data processing system using deep learning models, in particular a hybrid CNN+LSTM architecture.

During the qualification work, the structure of the data processing system was analyzed, time series based on activity logs were prepared, and seasonal and unstable data characteristics were identified. Several forecasting and anomaly classification models were implemented, including MA, LSTM, and CNN+LSTM. Experiments were conducted with different amounts of historical input days to optimize accuracy. A mechanism for detecting anomalies based on prediction error and confidence interval was developed. The results indicate the high efficiency of the hybrid approach for monitoring and assessing the state of information systems in near real-time.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – інтерфейс прикладного програмування (англ., Application Programming Interface)

AUC – площа під кривою (англ., Area Under the Curve)

CNN – згорткова нейронна мережа (англ., Convolutional Neural Network)

DL – глибоке навчання (англ., Deep Learning)

EDA – розвідницький аналіз даних (англ., Exploratory Data Analysis)

FPR – хибно позитивна частота (англ., False Positive Rate)

GPU – графічний процесор (англ., Graphics Processing Unit)

IIoT – промисловий інтернет речей (англ., Industrial Internet of Things)

LSTM – довготривала короткочасна пам’ять (англ., Long Short-Term Memory)

MA – ковзне середнє (англ., Moving Average)

MAE – середня абсолютна похибка (англ., Mean Absolute Error)

ML – машинне навчання (англ., Machine Learning)

PCI – довірчий інтервал за процентилем (англ., Percentile Confidence Interval)

RMSE – корінь середньоквадратичної похибки (англ., Root Mean Squared Error)

ROC – крива робочих характеристик приймача (англ., Receiver Operating Characteristic)

TPR – істинно позитивна частота (англ., True Positive Rate)

ВСТУП

У сучасних умовах стрімкого зростання обсягів цифрових даних, складності архітектур програмного забезпечення та зростання вимог до надійності інформаційних систем особливої актуальності набуває задача автоматизованого виявлення аномалій у роботі систем обробки даних. Аномалії можуть свідчити про збої, помилки в конфігурації, неочікувану поведінку користувачів або навіть кіберзагрози. Їх своєчасне виявлення дозволяє запобігти фінансовим збиткам, простою систем та втраті даних, а також є важливим етапом у забезпеченні стабільного функціонування цифрової інфраструктури.

Традиційні підходи до виявлення аномалій, такі як статистичні правила, порогове детектування або фіксовані евристики, виявилися недостатньо ефективними в умовах динамічного, нелінійного та багатовимірного середовища, де системи обробки даних постійно змінюються. У цьому контексті методи глибокого навчання, які здатні автоматично навчатися з даних, виявляти приховані закономірності та працювати з великими обсягами інформації, відкривають нові можливості для розв’язання задачі детекції аномалій.

Особливо ефективними виявилися такі архітектури, як автокодери (autoencoders), рекурентні нейронні мережі (RNN, LSTM), згорткові мережі (CNN), а також комбіновані гібридні моделі. Завдяки здатності до самонавчання без потреби в анотованих вибірках, ці моделі можуть бути адаптовані до широкого спектра задач – від аналізу часових рядів телеметрії до контролю стабільності кластерів у хмарних обчисленнях.

Метою даної кваліфікаційної роботи є розробка, реалізація та дослідження методу виявлення аномалій у роботі системи обробки даних із використанням моделей глибокого навчання. У роботі буде проведено аналіз існуючих підходів до детекції аномалій, розглянуто архітектури нейронних

мереж, адаптовані до обробки часових рядів, а також реалізовано програмний прототип системи, здатної автоматично ідентифікувати відхилення у функціонуванні обчислювальних сервісів. Ефективність розробленої моделі буде оцінена за допомогою метрик точності, повноти, F1-міри та середнього часу виявлення.

1 ВИЯВЛЕННЯ АНОМАЛІЙ

1.1 Вступ до задачі виявлення аномалій

У сучасному світі цифрової інформації обробка великих обсягів даних стала основою для прийняття рішень у багатьох галузях – від інформаційної безпеки та фінансового аналізу до охорони здоров’я та промислової автоматизації. Одним із критично важливих аспектів аналізу даних є виявлення аномалій – відхилень, які не вписуються в загальну закономірність. Розуміння та виявлення таких аномалій дозволяє не лише запобігати небажаним наслідкам (збоям, шахрайству, витокам даних), але й відкривати нові закономірності, які можуть мати як негативне, так і позитивне значення.

Аномалія у даних (англ. data anomaly, outlier, abnormality) – це таке спостереження або набір спостережень, які істотно відрізняються від решти даних у наборі. Залежно від контексту, це може означати:

- статистичне відхилення від очікуваного розподілу;
- порушення структури або взаємозв’язків у даних;
- виникнення нових, нетипових шаблонів поведінки.

У прикладних задачах, таких як моніторинг мережевого трафіку, трансакцій у банківській сфері, технічних параметрів у виробництві, виявлення аномалій є засобом своєчасного реагування на потенційно небезпечні ситуації.

Існує кілька класифікацій аномалій залежно від природи даних, характеру порушення закономірностей та задачі аналізу. Основні типи:

- точкові аномалії (Point Anomalies) Окреме значення суттєво відрізняється від решти. Наприклад, різке зростання температури датчика на одному з етапів виробництва;

- контекстуальні аномалії (Contextual Anomalies). Значення може бути нормальним у загальному, але нетиповим у певному контексті (час, локація тощо). Наприклад, висока температура тіла вночі у здорової людини – аномальна в контексті добового циклу;
- колективні аномалії (Collective Anomalies) Група спостережень разом утворює аномалію, хоча кожне з них окремо – нормальне. Наприклад, шаблон з декількох правильних дій, що в сукупності становлять шахрайську атаку.

Аномалії можуть бути як справжніми, тобто свідчити про реальні зміни або проблеми в системі, так і штучними, пов’язаними з помилками або шумом у даних.

Справжні причини:

- атаки (кібербезпека, вторгнення);
- поломки обладнання;
- зміни навколишнього середовища;
- шахрайство або зловживання;

Штучні причини:

- помилки вимірювання;
- програмні помилки;
- неправильне зчитування або введення даних;
- втрата даних або дублікація.

1.2 Методи виявлення аномалій

Існує кілька основних підходів до виявлення аномалій у даних.

Статистичні методи базуються на припущенні про нормальний розподіл даних. Значення, що виходять за межі стандартних відхилень, вважаються аномальними. Приклад: метод Z-оцінки, межі міжквартильного розмаху.

Статистичні методи є одними з найстаріших і найбільш інтерпретованих підходів до виявлення аномалій у даних. Основна ідея таких методів полягає в тому, що нормальні дані слідують певному розподілу (часто нормальному), тоді як аномалії суттєво відхиляються від цього розподілу. Одним з базових підходів є використання Z-оцінки (z-score), яка дозволяє визначати, наскільки далеко значення відхиляється від середнього. Якщо значення перевищує певний поріг (наприклад, $|z| > 3$), його можна вважати аномальним. Цей метод добре працює для одновимірних даних із нормальним розподілом.

Іншим підходом є використання методу скользного (ковзного) середнього та дисперсії, які широко застосовуються в часових рядах. Аномалії виявляються при різкій зміні середнього значення або стандартного відхилення в короткому часовому вікні порівняно з історичним контекстом. Такий метод особливо корисний для моніторингу динамічних процесів, наприклад, навантаження на сервери, трафіку в мережах чи температурних змін.

Більш складні статистичні підходи включають побудову моделей розподілу ймовірності, таких як гаусівські суміші (Gaussian Mixture Models, GMM), де кожне спостереження має ймовірність належати до певного кластера. Значення з низькою ймовірністю належності можуть вважатися аномальними. Також використовуються методи на основі щільності, як-от Kernel Density Estimation (KDE), де аномалії – це точки з низькою щільністю розподілу. Перевагою статистичних методів є їх простота та пояснюваність, однак вони мають обмеження: чутливі до припущень про розподіл даних і менш ефективні в умовах високої розмірності або складних структур аномалій. Тому часто вони застосовуються як базові або допоміжні методи в комбінованих системах виявлення аномалій.

Методи машинного навчання є потужним інструментом виявлення аномалій у складних, високовимірних і нерегулярних наборах даних. Основна перевага цих методів полягає у здатності адаптуватися до структури

даних і знаходити приховані закономірності без необхідності формального опису розподілу. Залежно від наявності міток у даних, методи машинного навчання поділяються на контрольовані (supervised), неконтрольовані (unsupervised) та напівконтрольовані (semi-supervised).

Контрольовані методи виявлення аномалій потребують попередньо позначених даних, де відомо, які приклади є нормальними, а які – аномальними. До таких методів належать класифікатори, як-от дерево рішень, логістична регресія, метод опорних векторів (SVM), а також ансамблеві підходи, зокрема Random Forest та XGBoost. Вони будують гіперплощину або логічну модель, що розділяє нормальні та аномальні точки. Проте їхнім обмеженням є потреба в достатній кількості міток аномалій, що в реальних задачах трапляється рідко.

Неконтрольовані методи не вимагають міток і використовуються, коли невідомо, які дані є аномальними. Найпоширенішим методом є кластеризація, наприклад, алгоритм K-середніх, де точки, що не належать до жодного кластера або знаходяться далеко від центрів кластерів, вважаються аномаліями. Іншим потужним методом є ізоляційний ліс (Isolation Forest), який ізолює точки за допомогою випадкових дерев: аномалії легше ізолювати, отже вони мають меншу довжину шляху. Метод локальної щільності LOF (Local Outlier Factor) оцінює аномальність на основі відстані до найближчих сусідів, виявляючи точки, що розташовані в областях з низькою щільністю.

Напівконтрольовані методи навчаються тільки на нормальних даних, а потім визначають відхилення від цієї моделі як аномалії. Прикладом є автоенкодери – нейронні мережі, які навчаються відтворювати вхідні дані з мінімальними втратами. Якщо на вхід подати аномальні дані, модель не зможе їх коректно реконструювати, і рівень похибки зросте. Також використовуються нейронні мережі типу LSTM для обробки часових рядів – вони здатні виявляти аномальні послідовності, які відрізняються від очікуваного патерну. Методи машинного навчання демонструють високу

гнучкість, масштабованість і здатність працювати з реальними складними даними, тому є особливо актуальними для виявлення аномалій у кібербезпеці, індустріальному моніторингу, фінансовому аналізі та системах Інтернету речей.

Методи глибокого навчання останнім часом набули великого поширення у задачах виявлення аномалій завдяки своїй здатності автоматично витягувати релевантні ознаки з високовимірних і неструктурованих даних, таких як зображення, відео, звукові сигнали, часові ряди та текст. Ці методи ґрунтуються на глибоких нейронних мережах, які складаються з кількох шарів обробки, що дозволяє моделі навчатися на різних рівнях абстракції. Особливо ефективними є глибокі архітектури в умовах великого обсягу даних, де класичні алгоритми не справляються або вимагають ручної інженерії ознак.

Одним із найпоширеніших підходів у виявленні аномалій за допомогою глибокого навчання є використання автоенкодерів – нейронних мереж, що навчаються стискати вхідні дані у латентний простір, а потім відновлювати їх з нього. Ідея полягає в тому, що модель, натренована лише на нормальних прикладах, не зможе точно відновити аномальні зразки, що відображається у високому значенні функції втрат (loss). Варіації автоенкодерів, як-от варіаційні автоенкодери (VAE), дозволяють враховувати стохастичну природу даних і задавати ймовірнісний простір ознак, що робить детекцію більш інформативною.

Інший потужний підхід – рекурентні нейронні мережі (RNN), особливо архітектури типу LSTM (Long Short-Term Memory), які використовуються для обробки послідовних даних та часових рядів. У задачах виявлення аномалій LSTM моделі навчаються прогнозувати наступні значення ряду або реконструювати вхідні послідовності. Значне відхилення між реальними й передбаченими значеннями сигналізує про можливу аномалію. Такі методи активно застосовуються в фінансовому аналізі, моніторингу стану обладнання та системах кібербезпеки.

Окрему нішу займають глибокі згорткові нейронні мережі (CNN), які ефективно працюють із просторовими даними, такими як зображення або сигнали. У контексті виявлення аномалій CNN дозволяють знаходити невідповідності візуальним шаблонам, що особливо актуально в системах безпеки, контролі якості та медичній діагностиці. Також варто згадати генеративно-змагальні мережі (GAN), які складаються з генератора та дискримінатора. У разі використання GAN для виявлення аномалій, генератор навчається створювати реалістичні зразки нормальних даних, а дискримінатор – розрізняти реальні та згенеровані. Після навчання аномальні зразки легко виявити, оскільки вони не можуть бути згенеровані й мають високі оцінки дискримінатора.

У порівнянні з класичними алгоритмами, методи глибокого навчання є більш ресурсоємними й вимагають значної кількості даних для тренування. Проте вони забезпечують вищу гнучкість, масштабованість та здатність працювати з неструктурованими вхідними даними. Це робить їх незамінними у сферах, де точність, адаптивність і здатність працювати в реальному часі мають критичне значення, зокрема в інтелектуальних системах моніторингу, автономних пристроях, захисті критичної інфраструктури та діагностиці несправностей.

1.3 Сфери застосування

Сучасні системи кібербезпеки стикаються з дедалі складнішими загрозами, які постійно змінюються й вдосконалюються. Традиційні підходи на основі сигнатур або заздалегідь відомих шаблонів стають недостатніми, оскільки не здатні виявляти нові, невідомі або нульові (zero-day) атаки. У цьому контексті методи виявлення аномалій стали ключовим інструментом для проактивного захисту інформаційних систем, оскільки дозволяють знаходити незвичну активність у мережі, поведінці користувачів або роботі додатків без необхідності знати точний шаблон атаки.

Одним з основних напрямів застосування виявлення аномалій є мережевий моніторинг трафіку. У цій сфері аналізується потік даних, що проходить через комутатори, маршрутизатори та мережеві адаптери, з метою виявлення підозрілих шаблонів – незвичних портів, невідомих IP-адрес, великого обсягу передачі або частоти запитів. Наприклад, аномалія може вказувати на DDoS-атаку, сканування портів або витік інформації. Використовуючи методи кластеризації, машинного або глибокого навчання, можна класифікувати трафік як нормальний або аномальний у реальному часі. Особливо ефективно показали себе автоенкодери, які виявляють порушення структури вхідних даних, або RNN-моделі, що враховують послідовність запитів у часі.

Іншим важливим застосуванням є виявлення внутрішніх загроз (insider threats). Користувачі з доступом до критичних систем можуть навмисно чи випадково завдати шкоди організації. Методи виявлення аномалій аналізують поведінкові шаблони – частоту входу в систему, час доступу, зміну налаштувань, обсяг переданих даних тощо. Відхилення від звичайної поведінки можуть сигналізувати про компрометацію облікового запису або зловмисні дії. Наприклад, якщо працівник починає масово завантажувати файли у незвичний час доби, це може бути ознакою крадіжки даних. У таких сценаріях широко застосовуються системи UBA (User Behavior Analytics), що базуються на машинному навчанні.

Виявлення шкідливого програмного забезпечення також є критичною сферою застосування. Завдяки методам глибокого навчання можливо виявляти нові варіанти вірусів, троянів, програм-шпигунів або руткітів, аналізуючи структуру виконуваних файлів, їхню поведінку в системі або шаблони викликів API. Наприклад, CNN можуть аналізувати двійкове представлення файлу як зображення, знаходячи характерні ознаки шкідливого ПЗ. GAN-моделі можуть створювати нові сценарії атак і тренувати класифікатори на даних з імітацій, що значно покращує виявлення zero-day атак.

Ще одним практичним прикладом є захист хмарної інфраструктури. У хмарних середовищах велика кількість віртуальних машин, мікросервісів і контейнерів створює складну, динамічну архітектуру, яка важко піддається класичному контролю. Аномальне навантаження на CPU, зміни в шаблонах API-запитів або нетипова комунікація між сервісами можуть сигналізувати про вторгнення або внутрішні збої. Використовуючи моделі глибокого навчання, можливо створювати адаптивні системи моніторингу, які навчаються на основі звичайної поведінки та здатні ідентифікувати навіть незначні порушення.

Окрему увагу заслуговує інтеграція виявлення аномалій у SIEM-системи (Security Information and Event Management). Вони акумулюють журнали подій з різних джерел (сервери, бази даних, пристрої доступу) й у реальному часі аналізують їх. Виявлення аномалій додає до таких систем інтелектуальні функції аналізу, виявляючи події, які не потрапили під звичайні правила. Наприклад, різка зміна в логінах, віддалений доступ із незвичної геолокації чи повторювані невдалі входи можуть бути проаналізовані і класифіковані як загроза, навіть якщо вони не є відомим шаблоном атаки.

Узагальнюючи, можна сказати, що застосування виявлення аномалій у сфері кібербезпеки дозволяє перейти від реактивного до проактивного захисту, де потенційні загрози виявляються до того, як вони завдали шкоди. Інтеграція методів глибокого навчання, адаптивного моделювання та аналізу поведінки робить такі системи універсальними та надзвичайно ефективними в умовах зростаючої кількості кібератак та складності IT-інфраструктури. Ці технології вже використовуються в банківських установах, державних органах, хмарних платформах і критичній інфраструктурі, поступово стаючи стандартом для сучасної кібербезпеки.

Фінансовий сектор є однією з найбільш вразливих до шахрайства, відмивання коштів, маніпуляцій з транзакціями та інших аномальних дій. Величезні обсяги фінансових даних, що генеруються щоденно – транзакції,

перекази, платежі, запити кредитів, купівлі цінних паперів тощо – створюють як складність, так і можливість для впровадження інтелектуальних систем моніторингу. У цьому контексті методи виявлення аномалій набувають особливої важливості, оскільки дозволяють виявити підозрілі транзакції, які не підпадають під заздалегідь визначені правила, але мають потенційну загрозу.

Одним із ключових напрямів застосування виявлення аномалій є боротьба з фінансовим шахрайством. Системи виявлення шахрайства працюють у режимі реального часу, аналізуючи такі характеристики транзакцій, як сума, місце проведення, частота, спосіб оплати, IP-адреса, пристрій користувача тощо. Якщо виявляється нетипова поведінка – наприклад, різке збільшення суми витрат або транзакція з геолокації, яка не притаманна користувачу – система може автоматично заблокувати операцію або позначити її для подальшої перевірки. На практиці широко використовуються алгоритми кластеризації (наприклад, DBSCAN, K-means), моделі машинного навчання (Decision Trees, Random Forest, XGBoost), а також глибокі нейронні мережі, які вчаться з даних історичних операцій і виявляють нелінійні патерни.

У сфері відмивання грошей (AML, Anti-Money Laundering), де шахрайські дії маскуються під звичайну фінансову активність, виявлення аномалій є одним з небагатьох ефективних інструментів. Традиційні rule-based системи не завжди встигають адаптуватися до нових схем, у той час як аномалії – такі як серія дрібних транзакцій, що згодом об'єднуються в одну велику, або багаторазові перекази через складну мережу рахунків — можуть бути ідентифіковані за допомогою графових моделей, автоенкодерів або рекурентних нейронних мереж (RNN). Багато фінансових установ формують повні транзакційні графи клієнтів і будують на їх основі поведінкові моделі, що дозволяють виявити відхилення від нормального шаблону.

Ще одним важливим аспектом є аналіз біржової активності, особливо для виявлення маніпуляцій на фондових ринках. Нестандартна поведінка

трейдерів, незвична активність у позабіржовому обігу, синхронні операції з високою швидкістю, які важко помітити аналітикам, можуть бути виявлені системами, що аналізують послідовність котирувань, таймінг угод і зміни обсягів торгів. Глибокі моделі, зокрема трансформери або CNN, можуть розглядати часові ряди як сигнали й фіксувати навіть незначні аномалії, які можуть свідчити про маніпуляції або інсайдерську торгівлю.

Крім того, виявлення аномалій активно використовується в управлінні кредитними ризиками. Система оцінювання кредитоспроможності клієнтів аналізує велику кількість параметрів: рівень доходів, історію виплат, поведінку на інших фінансових платформах, тощо. Якщо у звичних патернах користувача з'являється аномалія – наприклад, одночасне подання заявок на кредити в кількох установах, різке збільшення боргового навантаження, зміна адреси або пристрою – це може сигналізувати про можливу спробу шахрайства або компрометацію облікового запису. Сучасні моделі машинного навчання дозволяють виявляти навіть приховані закономірності у таких даних.

Окрему увагу заслуговує інтеграція систем виявлення аномалій у бізнес-аналітику банків та фінансових установ. Наприклад, різке зниження або зростання обсягів переказів у певному регіоні, поява нових фінансових інструментів з незвичною активністю, зміна середнього чека або часового патерну витрат може сигналізувати про зміну поведінки клієнтів або про вплив зовнішніх факторів, зокрема кібератак, соціального тиску чи внутрішніх порушень. Аналіз таких аномалій дає змогу виявити ризики на ранньому етапі та адаптувати бізнес-процеси.

Таким чином, системи виявлення аномалій у фінансовому моніторингу вже сьогодні є критичним компонентом цифрової безпеки й стабільності. Вони не лише забезпечують оперативне виявлення загроз, але й формують основи для прийняття стратегічних управлінських рішень. Із розвитком технологій зростає точність, адаптивність і швидкодія таких систем, що дозволяє фінансовим інституціям реагувати на нові виклики максимально

ефективно. Очікується, що найближчим часом виявлення аномалій на основі глибокого навчання стане стандартом у фінансовій сфері поряд із традиційним аудитом, юридичними перевірками та регуляторним контролем.

Промисловий інтернет речей (Industrial Internet of Things, IIoT) є однією з найдинамічніших сфер цифровізації сучасного виробництва, яка передбачає масове використання датчиків, контролерів, машин та обчислювальних вузлів, з'єднаних у єдину систему для моніторингу, управління й оптимізації промислових процесів. У таких масштабних і критичних середовищах навіть незначні відхилення можуть призвести до збоїв, аварій або зупинки виробництва. Саме тому виявлення аномалій у IIoT-системах є ключовим елементом забезпечення надійності, безпеки та економічної ефективності виробничих підприємств.

В основі промислових систем IIoT лежить обробка потоків даних від тисяч сенсорів, які фіксують температури, тиски, вібрації, витрати енергії, швидкості обертання, хімічні показники та інші параметри. Ці дані надходять у реальному часі та потребують постійного аналізу. Через складність фізичних процесів та наявність численних зовнішніх факторів, які впливають на роботу обладнання, описати всі можливі стани системи аналітичними моделями надзвичайно складно. У зв'язку з цим, алгоритми виявлення аномалій, що базуються на машинному та глибокому навчанні, набувають широкого застосування.

Одним з ключових завдань є попередження відмов устаткування (predictive maintenance), що передбачає виявлення відхилень у роботі агрегатів до моменту їх фізичного виходу з ладу. Наприклад, якщо система моніторингу виявляє незвичну вібрацію, зміну температурного режиму або зниження ефективності змащення, це може свідчити про знос підшипника, проблему в системі подачі енергії або іншу потенційну несправність. У таких випадках методи виявлення аномалій, зокрема на основі автокодерів, LSTM-мереж або One-Class SVM, дозволяють виявити проблему ще до того, як вона призведе до простою чи пошкодження обладнання.

Іншим важливим напрямком є контроль якості продукції, особливо в галузях, де точність є критичною: хімічна промисловість, фармацевтика, мікроелектроніка. Тут відхилення у значеннях параметрів (наприклад, концентрації речовин, температури реакцій, тривалості обробки) можуть сигналізувати про порушення технологічного процесу. У таких випадках класичні правила контролю якості часто не дають повної картини, і на допомогу приходять методи безнаглядного навчання – кластеризація, ізоляційні ліси (Isolation Forest), Gaussian Mixture Models – які можуть фіксувати рідкісні відхилення, що виникають унаслідок комбінацій численних факторів.

Крім цього, виявлення аномалій у IIoT допомагає в енергоменеджменті. У великих виробничих комплексах аномальне споживання енергії або тепла може свідчити про приховані втрати, неправильну експлуатацію устаткування чи навіть про хибні налаштування. Наприклад, якщо одна з печей постійно витрачає більше енергії, ніж інші аналогічні, це може бути сигналом до технічної перевірки. Методи аномального виявлення дозволяють автоматично знаходити такі відхилення та підвищувати енергоефективність.

Особливої актуальності набуває виявлення аномалій у IIoT у сфері інформаційної безпеки виробництва. З підключенням обладнання до Інтернету з’являється новий вектор атак – кіберзлочинці можуть втручатися у роботу пристроїв, змінювати параметри або зчитувати критичну інформацію. У цьому контексті виявлення аномалій у мережевому трафіку, несподівані команди до контролерів або нетипова частота запитів можуть свідчити про проникнення або спробу компрометації. Інструменти виявлення аномалій дозволяють виявити такі атаки на ранньому етапі та знизити ризики.

Ще один важливий аспект – адаптація систем моніторингу до змін у виробництві. У реальних умовах потужність ліній може змінюватися, вводяться нові рецептури або переоснащується обладнання. У таких умовах ручне оновлення всіх правил контролю стає непрактичним. Методи

аномального виявлення можуть працювати у режимі постійного навчання на поточних даних (continual learning), адаптуючи свої уявлення про «норму» до змін у процесах.

Таким чином, виявлення аномалій у промисловому інтернеті речей є не лише інструментом підвищення надійності та продуктивності, а й одним із ключових факторів розвитку концепції Індустрії 4.0. Завдяки йому підприємства можуть переходити від реактивної до проактивної моделі управління, мінімізуючи втрати, підвищуючи безпеку та ефективність. Із розвитком 5G, хмарних обчислень і edge-аналітики роль систем виявлення аномалій у IIoT буде лише зростати, охоплюючи дедалі ширше коло застосувань – від нафтопереробки до харчової промисловості.

Аномалії у даних – це важливе поняття, що лежить в основі багатьох систем автоматичного аналізу, діагностики, безпеки та контролю. Їхнє своєчасне виявлення є ключем до забезпечення надійності й стабільності складних інформаційних, технічних та економічних систем. Розвиток методів виявлення аномалій продовжує бути пріоритетним напрямком досліджень у галузі машинного навчання, статистики та інженерії даних.

2 АРХІТЕКТУРИ CNN І LSTM

2.1 Згорткові нейронні мережі

Згорткові нейронні мережі (CNN) – це нейронні мережі, які застосовуються в різних галузях, таких як класифікація зображень та розпізнавання обличчя [1]. Останнім часом CNN також використовуються для виявлення аномалій у часових рядах на основі прогнозування. Архітектура CNN складається з трьох основних компонентів: згорткового шару, шару pooling та повнозв’язного шару. Приклад простої CNN наведено на рисунку 2.1 [1].

Згортковий шар використовується для виділення ознак із вхідних даних шляхом переміщення фільтрів (ядер) певного розміру по виходу попереднього шару. Після згортки застосовуються функції активації для підвищення нелінійності [2]. Широко використовується функція активації ReLU, яка встановлює всі від’ємні значення у карті ознак в нуль.

CNN може складатися з кількох згорткових шарів, при цьому перший шар виділяє найпростіші ознаки, наприклад, краї. Наступні згорткові шари витягують дедалі більш абстрактні ознаки. Після кожної згортки знову використовується функція активації для підвищення нелінійності виходу.

Після згорткових шарів застосовуються pooling-шари, які зменшують розмір карт ознак і, відповідно, обчислювальну складність. Нарешті, повнозв’язний (dense) шар використовується для перетворення витягнутих ознак у вихід у формі класів, ймовірностей або числових значень [3].

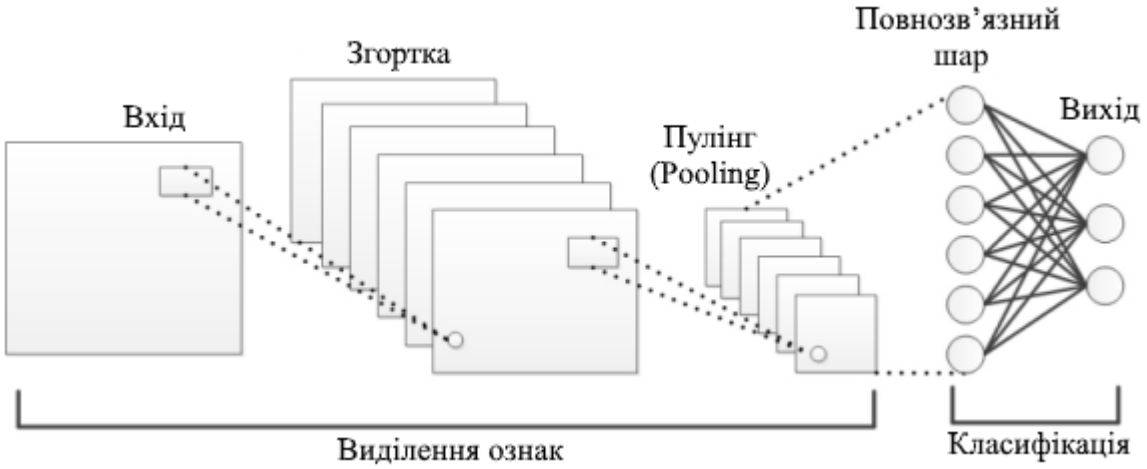

Рисунок 2.1 – Архітектура CNN

2.2 LSTM (Long Short-Term Memory)

Рекурентні нейронні мережі (RNN) не лише формують вихід на основі поточного стану, але й враховують попередній стан. Попри цю властивість, RNN стикаються з однією серйозною проблемою під час навчання – зникання градієнтів. Значення ваг у нейронній мережі оновлюються за допомогою градієнтів. Проте, коли градієнт поширюється назад у часі, його значення стає надзвичайно малим і майже не впливає на навчання. Особливо сильно це проявляється у початкових шарах RNN, де оновлення ваг стає надто слабким. Через це RNN не здатні зберігати інформацію протягом довгих послідовностей [3].

LSTM була розроблена для подолання проблеми зникання градієнтів. У мережі LSTM введено внутрішні петлі, які дозволяють відкидати непотрібну інформацію та зберігати важливу. LSTM використовує систему воріт (gating units): ворота забування (forget gate), вхідні ворота (input gate) та вихідні ворота (output gate). Ця система керує тим, яка інформація може пройти далі, а яка – ні. На рисунку 2.2 представлено схему, яка ілюструє стандартну архітектуру LSTM [4].

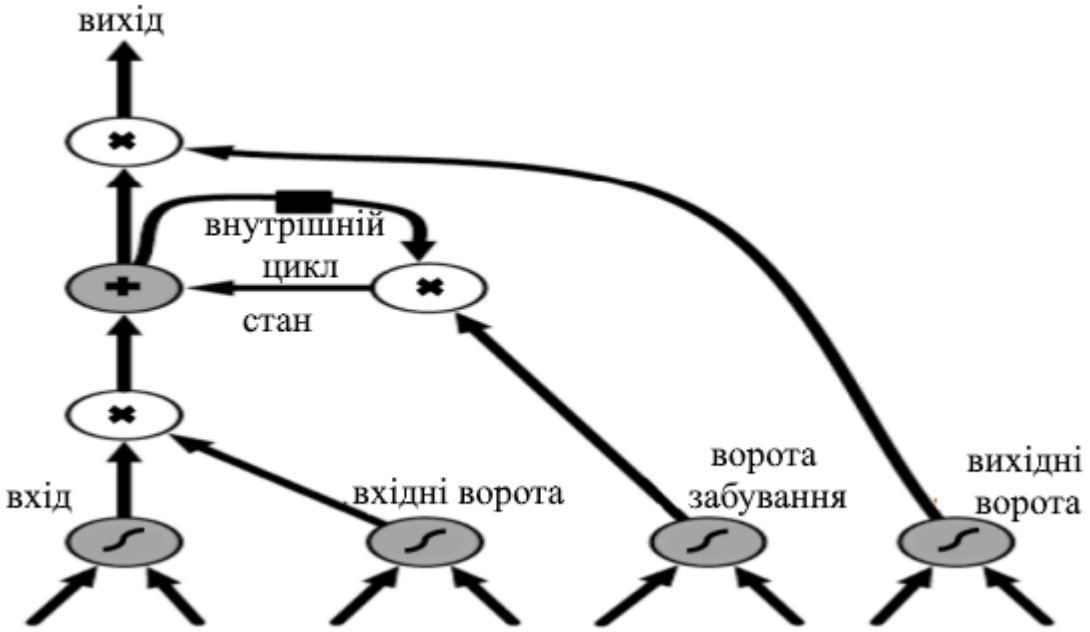

Рисунок 2.2 – Архітектура LSTM

Ворота забування (F_t, де t – поточний момент часу) керують потоком інформації зі стану комірки (C_t). Значення F_t визначається за допомогою сигмоїдної функції (σ) і вказує, яка кількість інформації буде пропущена: де 1 означає повне збереження інформації, а 0 – повне її видалення.

X_t, i_t, h_t та y_t відповідають, відповідно, за вхід, поточне значення на вході, поточну внутрішню інформацію та вихід комірки в момент часу t.

У всіх воротах ваги (W) та зсуви (b) оновлюються аналогічним чином, проте з використанням різних параметрів.

Вихідні ворота (o_t) визначають, яка інформація передається до наступного прихованого стану в комірці і. Нові елементи пам’яті c̃ створюються за допомогою гіперболічного тангенса, що обумовлено його швидкою обчислювальною ефективністю.

Позначення (·) вказує на поелементне множення двох векторів [5].

$$f_t = \sigma(W_{fh} h_{t-1} + W_{fx} x_t + b_f), \tag{2.1}$$

$$i_t = \sigma(W_{ih} h_{t-1} + W_{ix} x_t + b_i), \tag{2.2}$$

$$\tilde{c}_t = \tanh(W_{\tilde{c}h} h_{t-1} + W_{\tilde{c}x} x_t + b_{\tilde{c}}), \tag{2.3}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \tag{2.4}$$

$$o_t = \sigma(W_{oh} h_{t-1} + W_{ox} x_t + b_o), \tag{2.5}$$

$$h_t = o_t \cdot \tanh(c_t). \tag{2.6}$$

Крім того, у порівнянні зі звичайними рекурентними нейронними мережами (RNN), мережа LSTM найкраще підходить для навчання довгострокових залежностей та обробки часових рядів [6]. Архітектура LSTM та рівняння 2.1–2.6 також будуть використані в гібридній архітектурі CNN+LSTM.

2.3 Архітектура CNN+LSTM

Мережа CNN+LSTM – це гібридна модель, що поєднує як CNN, так і LSTM. CNN використовується для вилучення просторових ознак з вхідних даних за допомогою двох згорткових шарів, після яких іде шар пулінгу для зменшення кількості карт ознак та обчислювальних витрат. Отриманий результат передається до мережі LSTM, яка призначена для вивчення довгострокових залежностей у послідовностях часових рядів [7]. У фіналі вихід останнього шару LSTM передається до (dense) шарів, які формують прогноз у вигляді числових значень, що відображають активність підсистеми обробки даних.

Архітектура CNN+LSTM поєднує можливості згорткових нейронних мереж (Convolutional Neural Networks, CNN) та довготривалої короткочасної пам’яті (Long Short-Term Memory, LSTM) для ефективної обробки послідовних даних, таких як часові ряди чи телеметричні потоки. Така

гібридна модель дозволяє виявляти як просторові особливості вхідного сигналу, так і часові залежності в ньому.

Рисунок 2.3 – Архітектура CNN+LSTM

На початковому етапі модель приймає вхідні дані у вигляді багатовимірної матриці, де кожен рядок або вектор відповідає одному часовому кроку. Першим оброблювальним блоком є згорткова частина (CNN), що складається з одного або кількох згорткових шарів. Ці шари виконують операції згортки за допомогою фільтрів (ядер), які проходять по даних і виділяють локальні патерни або особливості. Кожен згортковий шар супроводжується функцією активації ReLU, яка додає нелінійність у модель. Після цього застосовується шар субдискретизації (max pooling), який зменшує розмірність ознак і знижує обчислювальні витрати, зберігаючи при цьому найінформативніші характеристики сигналу.

Далі вихідні ознаки згорткових шарів трансформуються за допомогою шару вирівнювання (flatten або reshape), щоб забезпечити відповідний формат даних для подачі до LSTM. Рекурентна частина архітектури (LSTM) складається з одного або кількох шарів, які аналізують часову послідовність витягнутих ознак. LSTM забезпечує здатність зберігати довготривалі залежності, що дозволяє виявляти зміни у поведінці системи або приховані тренди в даних.

Після завершення обробки часових залежностей вихід останнього LSTM-шару передається на один або кілька щільних (Dense) шарів. Ці шари відповідають за фінальну інтерпретацію вхідних ознак і генерують вихід моделі. Залежно від типу задачі, вихід може мати вигляд класу, вірогідності або числового значення. Наприклад, у задачах виявлення аномалій вихід

може представляти очікуване значення, яке порівнюється з реальним, або рівень ймовірності того, що поточний стан є аномальним.

Таким чином, архітектура CNN+LSTM поєднує в собі здатність ефективно виявляти локальні характеристики сигналів і зберігати часовий контекст, що робить її особливо корисною для завдань аналізу телеметрії, фінансових рядів, кібербезпеки та інших задач, де важливі як просторові, так і часові патерни.

3 ПЛАНУВАННЯ ЕКСПЕРИМЕНТУ

3.1 Опис набору даних

Набор даних було взято з відкритого джерела на kaggle – це надані організаціъ, яка пропонує хмарне рішення для підприємств з метою забезпечення обміну даними між їхніми операційними системами. Це хмарне рішення є системою обробки даних, що складається з чотирьох підсистем, кожна з яких відповідає за обробку певного типу даних між системами кінцевих користувачів. Дані отримані з журналів активності всіх підсистем, в яких зафіксовано час і дату виконання дії, а також зазначено, яка саме підсистема здійснила обробку.

Дії з обробки є HTTP-запитами до API цільової системи кінцевого користувача. Один запит може надіслати дані до відповідної системи або отримати, оновити чи видалити дані з неї через її API. Для передачі даних з однієї системи в іншу потрібно кілька дій обробки, оскільки дані спочатку потрібно отримати, а потім передати. Однак кожна окрема дія обробки журналюється й буде присутня в наборі даних незалежно від типу запиту, оскільки це не є важливим у межах цього дослідження.

Часові ряди охоплюють період з 26.04.2021 по 22.03.2022 з інтервалом в одну годину.

Окрім основного набору даних, що містить часові ряди, існують ще чотири окремі набори даних. Кожен із них відповідає окремій підсистемі та містить часові позначки, коли у відповідному часовому ряді було зафіксовано аномалію.

Сирі дані містять майже 4,5 мільйона рядків і 2 стовпці: call_date_time та category, які ідентифікують час і дату, коли відбулася дія обробки, та підсистему, відповідальну за цю дію, відповідно. Дії обробки також

називаються API-викликами, викликами, активністю системи або поведінкою системи – відповідно до контексту цієї роботи.

Для виявлення аномальної поведінки навчається прогностична модель, яка прогнозує наступний часовий крок, тобто наступну годину. Вихідний набір даних містить кілька викликів однієї підсистеми за один часовий крок. Моделювання активності або поведінки підсистеми щогодини виконується за допомогою функції groupby+size. Ця функція візуалізована на рисунку 3.1. Компонент groupby створює групи для кожної підсистеми по годинах, а компонент size підраховує кількість викликів у кожній групі та повертає це значення у новому стовпці size, залишаючи лише один рядок на групу.

Після застосування цієї функції залишається близько 6400 рядків, кожен з яких представляє кількість викликів, здійснених підсистемою за певну годину. Якщо підсистема не здійснювала жодних викликів у певний момент часу, значення в стовпці size, що ідентифікує активність, встановлюється як 0.

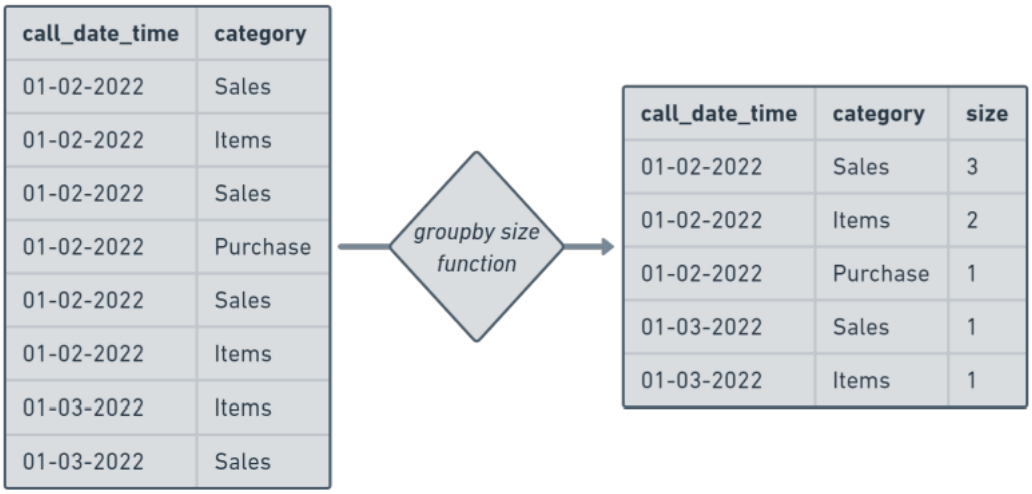

Рисунок 3.1 – Опис функції groupby + size

Після підготовки датасету, було проведено його попереднє дослідження з використанням графіків і описової статистики, таких як кількість спостережень, середнє значення, мінімальне та максимальне значення, стандартне відхилення і квартилі для кожної підсистеми. Часові

ряди було перевірено на стаціонарність і сезонність за допомогою графічного аналізу. Дані виявилися нестабільними (нестаціонарними), тобто середнє значення та дисперсія змінюються з часом.

Сезонність була візуалізована за допомогою діаграми boxplot на рисунку 3.2, де аналізувалася активність у різні дні тижня [8]. Два стовпчики праворуч демонструють найнижчий загальний рівень активності, який було отримано шляхом підсумовування кількості викликів усіх чотирьох підсистем. Це логічно пояснюється тим, що кінцевими користувачами здебільшого є організації, що значно активніші з понеділка по п’ятницю, ніж у вихідні дні.

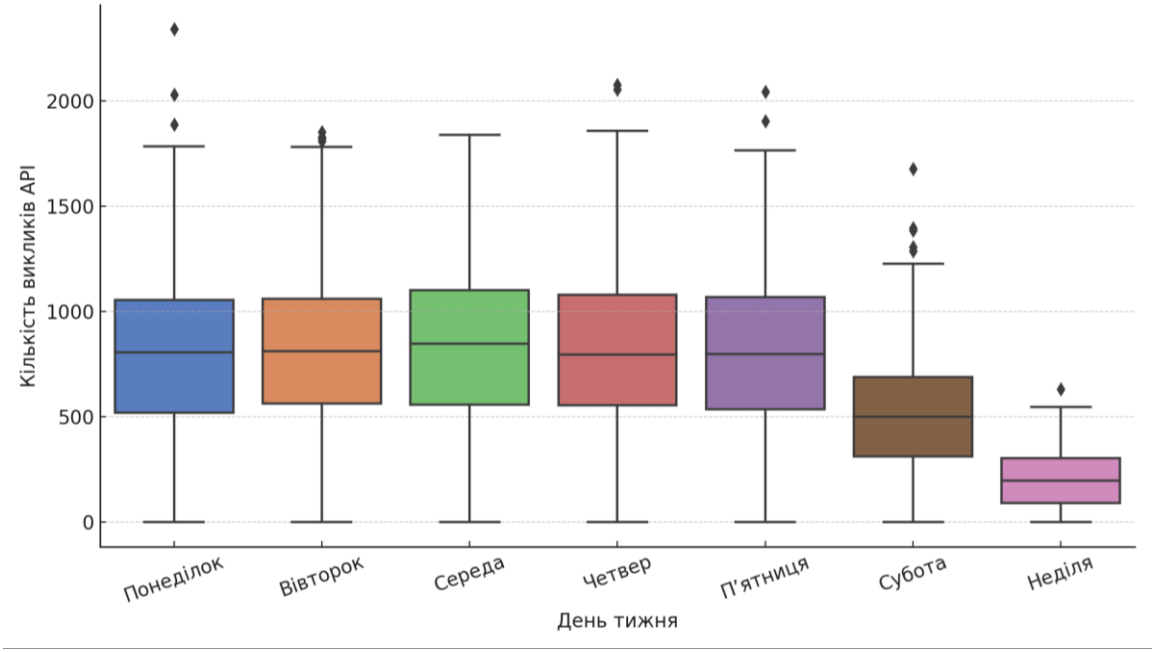

Рисунок 3.2 – Сезонності за тижнем

3.2 Дослідження даних

Для дослідження даних та отримання корисних інсайтів було проведено дослідницький аналіз даних (EDA) з використанням візуалізацій, таких як графіки та діаграми. Щоб отримати уявлення про поведінку кожної підсистеми з часом, було створено лінійний графік для кожної підсистеми за перший тиждень, як показано на рисунку 3.3.

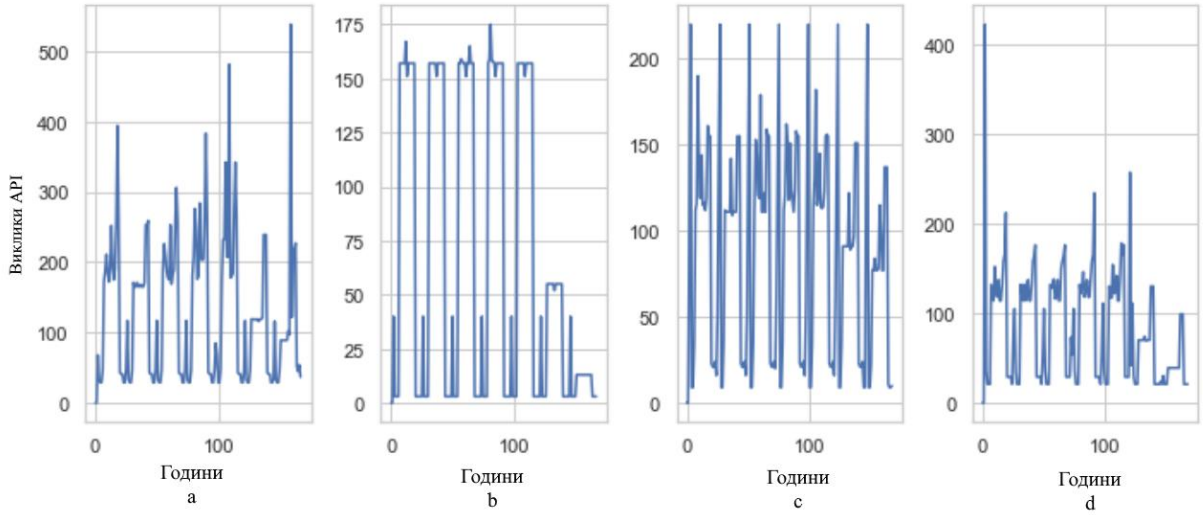

Рисунок 3.3 – Лінійні графіки активності підсистем за перший тиждень

Кожен графік представляє окрему підсистему, де a, b, c і d відповідають підсистемам Administrative, Items, Purchase та Sales відповідно. На графіках година 0 відповідає 26.04.2021 00:00, отже пік ліворуч відповідає понеділку, а праворуч – неділі.

Графік 3.3a ілюструє неочікувану поведінку викликів API з боку підсистеми Administrative з плином часу. Крім того, було виявлено сезонність, за якої суботи та неділі демонструють меншу активність порівняно з іншими днями. На графіку 3.3a чітко видно велику кількість API-викликів у неділю, що є рідкісним явищем, тому передбачається, що цьому передував якийсь нештатний випадок, який спричинив підвищену активність.

Далі, графіки також вказують на наявність певних спільних трендів між чотирма підсистемами протягом першого тижня. Наприклад, збільшення кількості викликів API в одній підсистемі відбувається одночасно зі збільшенням у іншій. Цей шаблон можна пояснити тим, що підсистеми певною мірою корельовані між собою – тобто зростання кількості викликів в одній підсистемі тягне за собою зростання в іншій.

Щоб перевірити це припущення та дослідити, чи існують інші кореляції між підсистемами, була побудована матриця кореляцій, що показана на рисунку 3.4. Графік кореляцій відображає силу зв’язку між двома

підсистемами: колір позначає величину кореляції – чим світліший колір, тим сильніший зв’язок. Загалом підсистеми мають відносно середньо-високу кореляцію між собою, середнє значення якої становить приблизно 0.709. Ця кореляція підтверджується графіком, оскільки на ньому спостерігаються різні відтінки.

Найсильніший зв’язок простежується між підсистемами Sales і Purchase. Це можна пояснити тим, що підсистема Sales обробляє дані, пов’язані з продажами, між системами кінцевих користувачів, тоді як підсистема Purchase – дані, пов’язані з закупівлями. Наприклад, якщо кінцевий користувач продає товар і реєструє замовлення на продаж у системі, це замовлення має бути оброблене іншою системою, і водночас може бути створене замовлення на закупівлю для поповнення запасів проданого товару.

Отже, з графіку кореляцій можна зробити висновок, що кореляції між підсистемами пов’язані з тригерами кожної системи: підсистеми, які викликаються іншими, матимуть сильну кореляцію, тоді як ті, що активуються внаслідок діяльності інших підсистем, демонструватимуть слабший зв’язок.

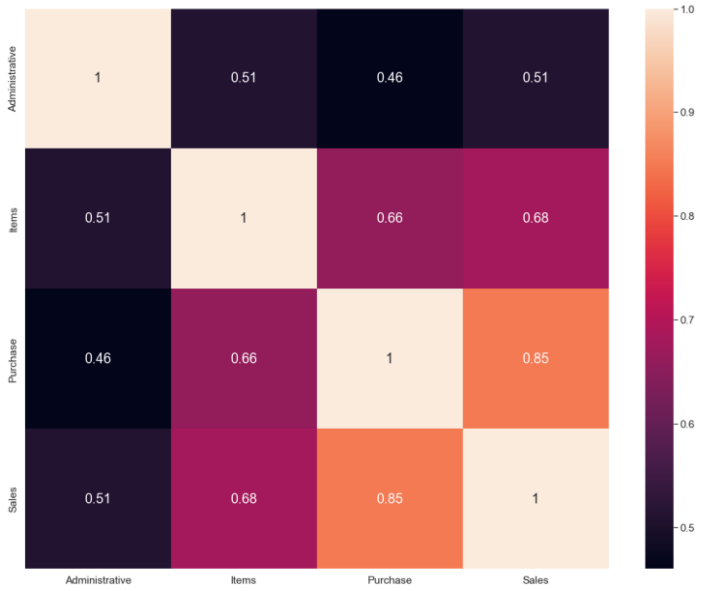

Рисунок 3.4 – Взаємозв'язок між підсистемами

3.3 Попередня обробка даних

Моделі глибокого навчання потребують послідовних (часових) даних для генерації точних прогнозів. Однак для полегшення інтерпретації даних моделями необхідне додаткове попереднє опрацювання.

Дані були розділені на три підмножини: навчальну, валідаційну та тестову. Розподіл здійснено у хронологічному порядку: навчальний набір містить найстаріші дані, тестовий – найновіші, а валідаційний охоплює період між ними. Часовий ряд починається з 26 квітня 2021 року о 00:00:00 і завершується 22 січня 2022 року о 23:00:00. Навчальний набір охоплює період від початку до 22 жовтня 2021 року 00:00, з якого починається валідаційний набір. Валідаційний набір триває до 12 грудня 2021 року 00:00 – моменту, з якого починається тестовий набір.

Призначення кожної з підмножин проілюстровано на рисунку 3.5. Як показано, навчальний набір використовується для навчання та налаштування початкових моделей, валідаційний – для оптимізації моделей шляхом підбору гіперпараметрів, а тестовий – для здійснення прогнозів та оцінки ефективності моделей.

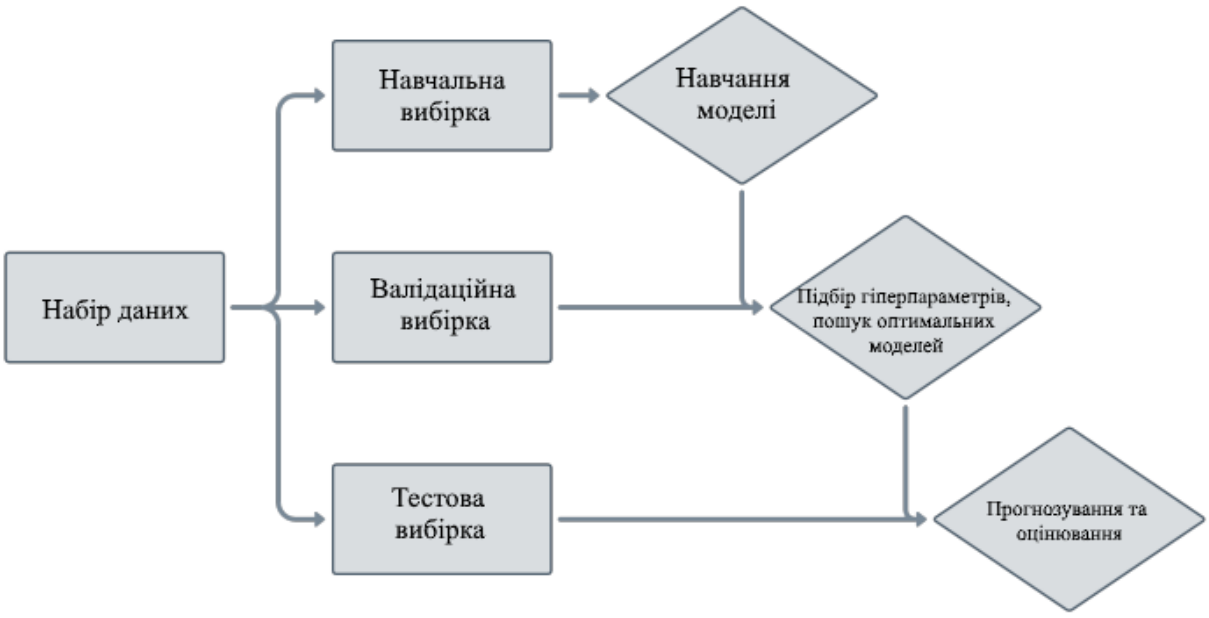

Рисунок 3.5 – Розподіл даних

Часові ряди були перетворені у часові послідовності, де за допомогою заздалегідь визначеного часово́го вікна (time window), яке задає кількість вхідних часових кроків, передбачається наступний часовий крок. Іншими словами, вхідні дані X містять дані за 1, 2, 5 або 7 днів, і на їх основі модель прогнозує значення y, яке відповідає наступному часовому кроку.

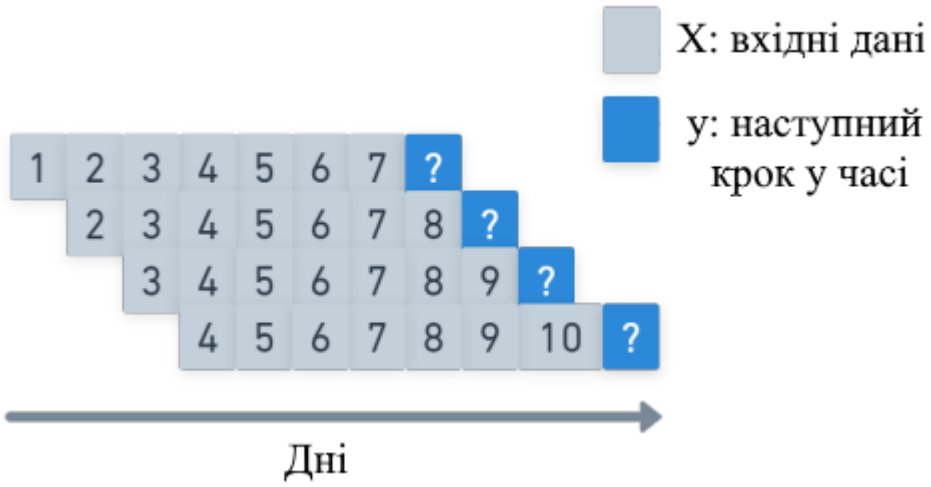

Рисунок 3.6 – Розподіл даних

Таблиця 3.1 ілюструє, як саме відбувається трансформація даних: наведено перетворення перших трьох послідовностей. Щоб краще зрозуміти, наведемо приклад: якщо вхідні дані починаються з дня 1 о 00:00 і закінчуються днем 7 о 24:00, то відповідне значення y (тобто прогноз) стосується дня 8 о 01:00.

3.4 Конвеєр дослідження

Підхід до виявлення аномалій на основі прогнозування часових рядів пояснюється за допомогою рисунка 3.7. Насамперед, дані часових рядів необхідно перетворити відповідно до заданого вікна часу, як було описано в попередньому розділі. Отримані послідовні дані використовуються як вхідні для моделей прогнозування, які генерують передбачення наступного кроку, як описано раніше.

Для кожного передбаченого значення обчислюється похибка прогнозу. Це значення похибки відповідає різниці між передбаченим значенням і фактичним значенням.

Далі розраховується довірчий інтервал за процентилем (PCI) для кожної підсистеми з використанням 97.5% довірчого рівня. Похибка прогнозу e порівнюється зі значенням PCI: якщо e перевищує PCI, така точка вважається аномальною; якщо ж e менше або дорівнює PCI, точка вважається нормальною.

Таблиця 3.1 – Перетворення часових рядів на послідовні дані з часовим вікном у 7 вхідних днів

дні дані (X)	Вихід (Прогноз y)
День 1–7 7 днів = 168 годин	День 8, перша година 169-й часовий крок
День 2–8 7 днів = 168 годин	День 9, перша година 337-й часовий крок
День 3–9 7 днів = 168 годин	День 10, перша година 504-й часовий крок

Набір даних містить одну числову змінну – цільову змінну size, яка варіюється від мінімального значення 0 до максимального 9084. Такий великий діапазон значень зумовлює високу дисперсію, що викликає необхідність нормалізації даних. Нормалізація забезпечує приведення всіх значень до заданого діапазону, що дозволяє уникнути впливу різних масштабів значень.

Для нормалізації викликів API було обрано MinMaxScaler з бібліотеки sklearn, оскільки саме цей підхід часто використовується у пов’язаних дослідженнях.

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \qquad 3.1$$

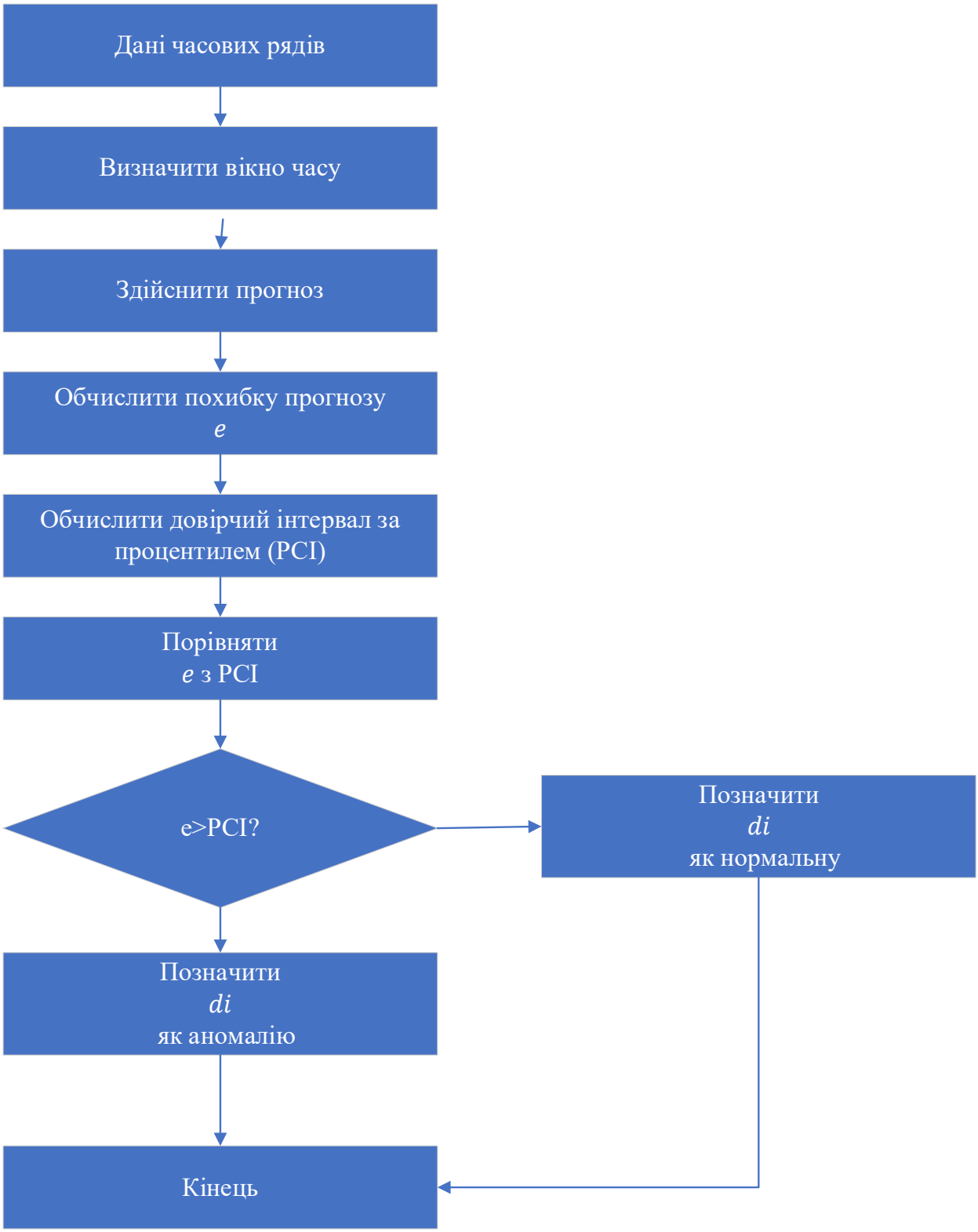

Рисунок 3.7 – Виявлення аномалій на основі прогнозування за даними часових рядів

Функцією MinMaxScaler спочатку було підігнано до тренувального набору даних, щоб навчитися на значеннях, що трапляються в цій підмножині. Після цього підігнаний масштабувач використовувався для перетворення значень змінної size, яка містить кількість викликів API, у тренувальному, валідаційному та тестовому наборах. Таким чином, усі значення в підмножинах були трансформовані на основі параметрів, вивчених при навчанні на тренувальному наборі. Це потрібно робити саме так, щоб уникнути витоку інформації – якби масштабувач навчався на валідаційному або тестовому наборі, він отримав би доступ до даних, які ще не повинен бачити.

Попередньо заданий діапазон значень встановлено від 0 до 1, оскільки неможливо мати від'ємну кількість викликів API. На етапі прогнозування значення зворотно масштабуються, щоб перетворити нормалізовані значення назад у початкові значення в діапазоні від 0 до нескінченності, що робить результати більш інтерпретованими.

Експериментальна фаза складається з двох основних компонентів: прогнозування часового ряду на один крок вперед і виявлення аномалій. Перший компонент оцінюється за допомогою метрик середньої абсолютної похибки (MAE) і кореня середньоквадратичної похибки (RMSE). Ці метрики були визнані придатними для оцінки точності прогнозування часового ряду в попередніх дослідженнях.

Другий компонент оцінюється за допомогою площі під кривою (AUC), показника істинно позитивних спрацьовувань (TPR), показника хибно позитивних спрацьовувань (FPR), точності (accuracy) і кривої робочих характеристик приймача (ROC-кривої). Ці метрики широко використовуються в суміжних дослідженнях для оцінювання ефективності моделей у класифікації аномалій.

Моделі глибокого навчання порівнюються з базовою (еталонною) моделлю, яка слугує точкою відліку. У задачах прогнозування часового ряду часто використовують наївні моделі, такі як ARIMA, AR і MA.

У цьому дослідженні як базову модель обрано MA (модель ковзного середнього) через її простоту реалізації та інтерпретації. MA обчислює середнє значення у межах ковзного вікна. Якщо ковзне вікно встановлено на 7 днів, воно використовуватиме середню кількість викликів API за дні з 1 по 7 як прогноз на наступний крок. Наступне передбачене значення буде середнім за дні з 2 по 8 і так далі. Як показано на рисунку, X – це вхідні дані, що складаються із 7 днів, а y – це результат, тобто середнє цих даних.

Підбір гіперпараметрів було використано для знаходження оптимальних моделей прогнозування LSTM та CNN+LSTM для одноетапного прогнозування поведінки системи обробки даних. Щоб досягти балансу між якісним навчанням моделі та її здатністю до узагальнення, для вибору моделей і гіперпараметрів, що використовуватимуться для передбачень і оцінки на тестовому наборі, застосовувались значення MAE на валідаційному наборі.

Існує велика кількість гіперпараметрів, які можна налаштовувати як для LSTM, так і для CNN+LSTM: кількість шарів, розмір пакета (batch size), кількість епох, оптимізатор, функція активації, функція втрат, кількість нейронів у шарі, dropout та регуляризація. Враховуючи обмежений час та обчислювальні ресурси, для налаштування було обрано три гіперпараметри: розмір пакета, кількість епох і оптимізатор.

Різниця між різними batch size та кількістю епох була незначною, тому було обрано batch size 100 і 50 та 30 епох для моделей LSTM і CNN+LSTM відповідно. Щодо оптимізатора, то результати відрізнялися суттєво, і найкращі результати за мінімальним значенням помилки показав оптимізатор Adam — як для LSTM, так і для CNN+LSTM.

На завершення було обрано відповідну функцію активації — ReLU, яка в попередньому аналізі показала найкращу продуктивність у прогнозуванні.

Огляд оптимальних архітектур моделей LSTM та CNN+LSTM після налаштування гіперпараметрів наведено у таблицях 3.2 та 3.3 відповідно.

Таблиця 3.2 – Модель LSTM

Шар (тип)	Форма виходу	Кількість параметрів
LSTM	(None, 100)	42 000
RepeatVector	(None, 1, 100)	0
LSTM	(None, 1, 100)	80 400
TimeDistributed	(None, 1, 4)	404

Таблиця 3.3 – Модель CNN+LSTM

Шар (тип)	Форма виходу	Кількість параметрів
Conv1D	(None, 22, 64)	832
Conv1D	(None, 20, 64)	12 352
MaxPooling1D	(None, 10, 64)	0
Flatten	(None, 640)	0
RepeatVector	(None, 1, 640)	0
LSTM	(None, 1, 150)	474 600
TimeDistributed	(None, 1, 100)	15 100
TimeDistributed	(None, 1, 4)	404

Програмне забезпечення, використане для реалізації цього дослідження – Python.

4 РЕЗУЛЬТАТИ ЕКСПЕРЕМЕНТУ

4.1 Прогнозування часових рядів

Моделі були навчені для прогнозування наступного кроку часу на основі історичних вхідних даних. Архітектури LSTM та CNN+LSTM порівнюються з базовою моделлю MA шляхом обчислення оціночних метрик MAE (середня абсолютна помилка) та RMSE (корінь середньоквадратичної помилки). Значення MAE та RMSE обчислюються окремо для кожної підсистеми, а також виводиться усереднений результат для всіх моделей. Помилки по підсистемах розраховані з аналітичною та відтворюваною метою, що дозволить у майбутніх дослідженнях додавати нові системи як ознаки до датасету без втрати можливості оцінювання на рівні окремих підсистем.

У таблицях 4.1 та 4.2 наведено порівняння продуктивності моделей LSTM і CNN+LSTM із базовою моделлю. Результати показують, що використання LSTM не покращує продуктивність порівняно з MA. Натомість, модель CNN+LSTM демонструє суттєве покращення результатів порівняно з базовою моделлю MA.

Водночас, продуктивність CNN+LSTM знижується при збільшенні кількості вхідних днів, у той час як модель LSTM показує нестабільну динаміку.

Таблиця 4.1 – Середнє значення MAE для 4 підсистем з історичним введенням за 1, 2, 5, 7 днів.

Метод	1 день	2 дні	5 днів	7 днів
MA	0.1618	0.1711	0.1807	0.1836
LSTM	0.1483	0.1501	0.1250	0.1296
CNN + LSTM	0.0949	0.1056	0.1241	0.1305

Таблиця 4.2 – Середнє значення RMSE для 4 підсистем з історичним введенням за 1, 2, 5, 7 днів.

Метод	1 день	2 дні	5 днів	7 днів
MA	0.2196	0.2323	0.2432	0.2485
LSTM	0.3482	0.2659	0.2173	0.2412
CNN + LSTM	0.1824	0.1967	0.2211	0.2329

4.2 Виявлення аномалій

Основною метою цього дослідження є виявлення аномалій у поведінці системи обробки даних. Для оцінювання ефективності класифікації аномалій використовувалися такі метрики: AUC, TPR, FPR, accuracy та ROC-крива. Результати представлені в таблиці 4.3 і демонструють ефективність кожної моделі.

Таблиця 4.3 – Оцінок ефективності

Метод	AUC	TPR (Чутливість)	FPR (Хибнопозитивна частка)	Точність (Accuracy)
MA	0.7639	0.5341	0.0198	0.9762
LSTM	0.7844	0.5215	0.0183	0.9774
CNN + LSTM	0.9046	0.7996	0.0178	0.9804

Як видно з таблиці, модель LSTM дещо перевершує MA, але ця перевага не є суттєвою. Натомість модель CNN+LSTM значно перевершує як базову модель MA, так і LSTM, забезпечивши приріст AUC на 15,55% і 15,3% відповідно до MA та LSTM. Ці результати підтверджуються іншими

метриками: TPR, FPR та accuracy також показали значне покращення для моделі CNN+LSTM порівняно з іншими моделями.

ROC-криві для кожної моделі зображено на рисунку 4.1, що візуалізує порівняння між показником істинно позитивних спрацювань (TPR) та хибнопозитивних спрацювань (FPR). Криві ROC також показують незначне покращення моделі LSTM у порівнянні з MA, проте воно не є суттєвим. З іншого боку, можна інтерпретувати, що CNN+LSTM перевершує як LSTM, так і MA з точки зору співвідношення TPR до FPR.

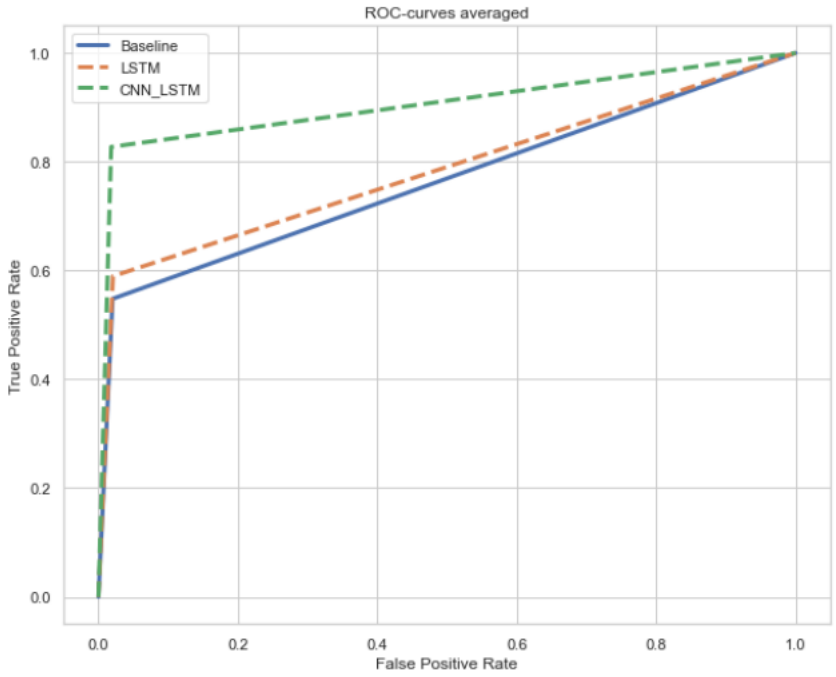

Рисунок 4.1 – ROC-криві для всіх моделей

Друге дослідження стосувалося вивчення впливу кількості вхідних днів, тобто обсягу історичних даних, на продуктивність моделей прогнозування. Таблиці 3 та 4 чітко демонструють покращення результатів моделі LSTM зі збільшенням кількості вхідних днів, однак, при подачі більше ніж 5 днів, спостерігається зниження її ефективності. Продуктивність LSTM покращується на 15,7% за MAE та на 37,5% за RMSE при збільшенні вхідних днів з 1 до 5. Суттєва різниця між значеннями MAE і RMSE свідчить

про наявність кількох великих помилок прогнозування, оскільки RMSE чутливіший до великих відхилень, ніж MAE.

З іншого боку, як модель MA, так і CNN+LSTM показують найкращі результати при використанні лише 1 вхідного дня. Якщо моделі отримують більше одного дня вхідних даних, ефективність CNN+LSTM суттєво знижується, тоді як MA демонструє лише незначне погіршення. Зниження ефективності моделі CNN+LSTM становить 37,5% за MAE та 27,7% за RMSE при збільшенні кількості вхідних днів з 1 до 7. Для порівняння, це ж саме зниження для моделі MA становить лише 13,5% і 13,2% відповідно. Вплив кількості вхідних днів візуалізовано на графіках значень MAE і RMSE для різних розмірів часових вікон, що представлені на рисунках 4.2 та 4.3.

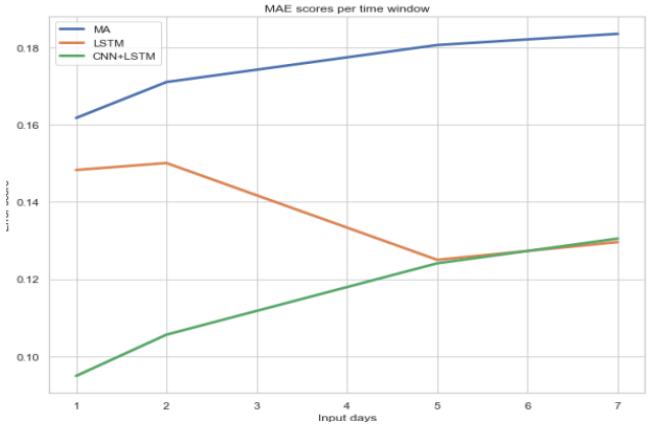

Рисунок 4.2 – Середні значення MAE при різній кількості вхідних днів

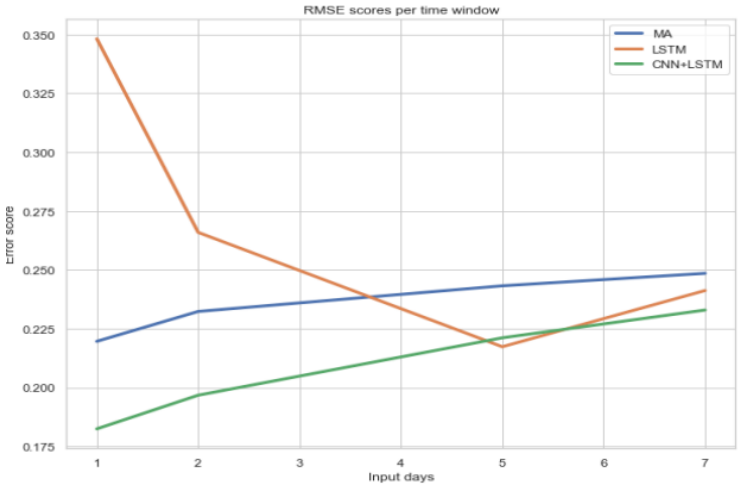

Рисунок 4.3 – Середні значення RMSE при різній кількості вхідних днів

ВИСНОВКИ

У межах цієї кваліфікаційної роботи було досліджено та реалізовано метод виявлення аномалій у поведінці системи обробки даних на основі гібридної моделі глибокого навчання CNN+LSTM. У процесі дослідження були виконані такі основні завдання:

- проведено аналіз особливостей функціонування системи обробки даних та її підсистем;
- підготовлено часові ряди на основі логів активності з урахуванням сезонності та нестаціонарності;
- побудовано та порівняно кілька моделей прогнозування (ma, lstm, cnn+lstm);
- проведено експериментальні дослідження щодо впливу кількості вхідних днів на точність прогнозування;
- реалізовано механізм виявлення аномалій на основі похибки передбачення та довірчих інтервалів;
- виконано оцінку моделей за метриками mae, rmse, auc, tpr, fpr та побудовано roc-криві.

На основі отриманих результатів можна зробити такі висновки:

1. Модель CNN+LSTM показала найкращі результати серед усіх розглянутих моделей як у задачі прогнозування, так і в задачі класифікації аномалій. Її використання дозволило підвищити точність виявлення відхилень у поведінці системи.

2. Вплив кількості історичних вхідних даних був різним для різних моделей: для LSTM спостерігалося покращення до певного моменту (5 днів), після чого точність знижувалась. У CNN+LSTM найкращі результати спостерігались при використанні одного дня як входу.

3. Методика виявлення аномалій на основі похибки прогнозу в поєднанні з довірчим інтервалом продемонструвала практичну ефективність і

дозволяє виявляти потенційно аномальні стани системи в режимі, близькому до реального часу.

4. Гібридна модель з використанням CNN для вилучення просторових ознак і LSTM для навчання на часових залежностях виявилась найбільш збалансованою за критеріями точності, стабільності та масштабованості.

5. Запропоноване рішення є придатним для подальшого впровадження у промислових системах моніторингу та забезпечення надійності IT-інфраструктур.

У майбутньому доцільно дослідити адаптивні механізми самонавчання моделі, а також розширити функціонал системи, включивши інші типи вхідних ознак, що відображають стан системи у більш широкому контексті.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ElSayed, Mahmoud Said, et al. "A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique." *Journal of Network and Computer Applications* 191 (2021): 103160.

2. Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., ... & Zhang, Q. (2020, November). Multivariate time-series anomaly detection via graph attention network. In 2020 IEEE international conference on data mining (ICDM) (pp. 841-850). IEEE.

3. Abdallah, Mahmoud Said ElSayed. Effective deep learning based methods for the anomaly detection in software-defined networks. Diss. University College Dublin. School of Computer Science, 2022.

4. Kelleher, John D. Deep learning. MIT press, 2019.

5. Пелещак, І., & Футрик, Ю. (2025). Прогнозування часових рядів за допомогою нейромережі з послідовно з'єднаними lstm блоками. Herald of Khmelnytskyi National University. Technical sciences, 347(1), 432-441.

6. Manaswi, N. K., Manaswi, N. K., & John, S. (2018). *Deep learning with applications using python* (Vol. 1, No. 45-56, p. 228). Berkeley, CA, USA: Apress.

7. Liu, Fan, Xingshe Zhou, Jinli Cao, Zhu Wang, Tianben Wang, Hua Wang, and Yanchun Zhang. 2022. Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional lstm-cnn. IEEE Transactions on Knowledge and Data Engineering, 34(6):2626–2640.

8. Pajankar, A. (2021). Introduction to data visualization with seaborn. In *Hands-on Matplotlib: Learn Plotting and Visualizations with Python 3* (pp. 243-267). Berkeley, CA: Apress.

9. Шандиба А. С., Ніколаєв О.Є Сітніков В. І Нанесення цифрових водяних знаків з використанням хаотичних карт *Системи управління, навігації та зв'язку*. 2024. № 4 с. 208-214