№ 1(56) 2026

МЕТОДИ ТА ПРИЛАДИ КОНТРОЛЮ ЯКОСТІ

ISSN 1993-9981 print
ISSN 2415-3575 online

https://mpky.nung.edu.ua

Прийнято 19.04.2026. Прорецензовано 08.05.2026. Опубліковано 30.05.2026.

УДК 697.85

DOI: 10.31471/1993-9981-2026-1(56)-237-249

ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНА СИСТЕМА ОПЕРАТИВНОГО СТАНУ ЯКОСТІ ВОДИ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Демчина М. М.
Кандидат технічних наук, доцент
Університет Короля Данила
76018, вул. Коновальця, 35, м. Івано-Франківськ, Україна
https://orcid.org/0009-0002-9161-4843
e-mail: mykolademchyna@gmail.com

Анотація. Протягом останніх десятиліть відбулося суттєве вдосконалення методів чисельного моделювання, що використовуються для прогнозування екологічних процесів у природних системах. Такий прогрес став можливим завдяки інтенсивному розвитку обчислювальної техніки, підвищенню точності математичних моделей та активному впровадженню сучасних методів аналізу даних у наукові дослідження. Сьогодні технології машинного навчання посідають важливе місце серед інструментів екологічного моніторингу, забезпечуючи нові можливості для обробки та інтерпретації великих масивів інформації про стан навколишнього середовища. Застосування інтелектуальних алгоритмів дає змогу не лише автоматизувати процес збору та аналізу даних, а й виявляти приховані взаємозв'язки між екологічними показниками. Це сприяє більш глибокому дослідженню складних природних систем, які характеризуються високою динамічністю та багатофакторністю. Особливого значення набуває можливість прогнозування кліматичних змін, а також аналізу гідрометеорологічних і біометричних параметрів, що дозволяє отримувати об'єктивну оцінку стану довкілля та формувати науково обґрунтовані управлінські рішення. Поєднання сучасних гідродинамічних моделей із методами інтелектуальної обробки даних сприяє підвищенню достовірності прогнозів та зменшенню рівня невизначеності під час моделювання екологічних процесів. Важливу роль у цьому відіграють супутникові системи спостереження, автоматизовані пости контролю та мережі сенсорів, які забезпечують накопичення комплексної інформації про циркуляцію водних мас, міграцію забруднювальних речовин, температурні зміни та інші процеси, що визначають екологічний стан водних об'єктів. Отже, інтеграція традиційних підходів математичного моделювання із сучасними технологіями аналізу даних створює ефективне підґрунтя для розроблення високоточних систем екологічного моніторингу. Подальший розвиток цього напряму, збільшення обсягів доступної інформації та розширення міждисциплінарної співпраці сприятимуть глибшому розумінню природних процесів і підвищенню ефективності природоохоронної діяльності на локальному, регіональному та державному рівнях.

Ключові слова: машинне навчання, водне середовище, екологія, прогнозування, інформаційно-вимірювальна система.

Запропоноване посилання: Демчина, М. М. (2026). Інформаційно-вимірювальна система оперативного стану якості води на основі машинного навчання. Методи та прилади контролю якості, 1(56), 237-249. doi: 10.31471/1993-9981-2026-1(56)-237-249

* Відповідальний автор

Вступ

У сучасних умовах збереження якості питної води є одним із ключових викликів для суспільства. Забезпечення населення безпечними водними ресурсами потребує постійного контролю та оцінювання основних показників якості води. Вода відіграє визначальну роль у життєдіяльності людини, а її стан безпосередньо впливає на здоров'я населення, рівень життя та соціально-економічний розвиток. Водночас посилення антропогенного навантаження, розвиток промислового виробництва, урбанізація та наслідки воєнних дій призводять до зростання ризиків забруднення водних об'єктів.

Оцінювання якості питної води передбачає комплексне дослідження її фізичних і хімічних характеристик. Особливу увагу приділяють виявленню небезпечних домішок, серед яких патогенні мікроорганізми, бактерії, сполуки важких металів, надлишковий вміст солей, залишки хлорвмісних речовин, різноманітні хімічні забруднювачі та механічні частинки. Значна частина таких речовин може потрапляти до водного середовища внаслідок скидання промислових стоків, аварійних викидів або інших видів господарської діяльності [1].

Застосування інформаційно-вимірювальних систем (ІВС) створює можливість підвищити ефективність контролю стану водного середовища, забезпечити більш точне визначення його параметрів та своєчасно виявляти негативні зміни. Це дозволяє оперативно реагувати на потенційні джерела забруднення та мінімізувати їхній вплив на екосистеми й населення.

Принцип функціонування сучасної інформаційно-вимірювальної системи ґрунтується на використанні ультразвукових методів контролю, які базуються на взаємодії ультразвукових хвиль із водним середовищем, а також на аналізі параметрів багаторазово відбитих сигналів. Поєднання таких підходів із сучасними засобами обробки даних забезпечує підвищення точності вимірювань, покращення чутливості системи та розширення її функціональних можливостей у задачах екологічного моніторингу [2].

Поєднавши ІВС з можливостями машинного навчання ми можемо отримати інструмент який дасть нам можливість точніше вимірювати параметри навколишнього середовища та швидше реагувати на можливі чинники забруднення завдяки прогнозування змін у вимірюваних параметрах.

Мета роботи – розробка та дослідження моделі машинного навчання на основі нейронних мереж для інтеграції в інформаційно-вимірювальні системи, що забезпечує оперативний моніторинг, аналіз кореляційних зв'язків та високоточне прогнозування показників якості й придатності водного середовища.

Аналіз сучасних закордонних і вітчизняних досліджень та публікацій

Контроль стану водних об'єктів та прогнозування змін екологічних показників є критично важливою технологією для забезпечення санітарно-епідеміологічної безпеки населення та сталого розвитку суспільства. Цей процес дедалі більше базується на впровадженні методів штучного інтелекту, комп'ютерного моделювання та автоматизованого аналізу великих масивів даних, і його використання в задачах екологічного моніторингу постійно розширюється.

У сучасній науковій літературі приділяється значна увага дослідженням факторів, які впливають на якість води на всіх етапах її підготовки та транспортування до споживача. Вітчизняні та закордонні дослідники відзначають, що традиційні технології знезараження (зокрема хлоруванні) та застарілі водоочисні споруди не завжди мають достатню ефективність і спроможність повністю усунути патогенні мікроорганізми чи небезпечні забруднювачі [3]. Як наслідок, активний розвиток отримали інноваційні методи фільтрації із застосуванням високоефективних сорбційних матеріалів, мембранних технологій та інноваційних способів хімічного очищення, які впроваджуються як на великих промислових об'єктах, так і в локальних системах водопідготовки [4].

В основі даної роботи лежить використання технологій машинного навчання та нейронних мереж для інтелектуального аналізу даних, оцінки кореляційних зв'язків між фізико-хімічними параметрами та оперативного прогнозування придатності води до споживання в інформаційно-вимірювальних системах.

Виклад основного матеріалу

Інструментарієм для реалізації задач машинного навчання однією з найпоширеніших мов програмування є Python. Її популярність зумовлена простотою синтаксису, високою читабельністю коду та широкими можливостями для розроблення програмних рішень різного рівня складності. Python підтримує сучасні підходи до об’єктно-орієнтованого програмування, містить ефективні структури даних і забезпечує швидке створення та тестування програмних продуктів.

Завдяки інтерпретованій архітектурі та великій кількості спеціалізованих бібліотек Python став одним із провідних інструментів у галузі штучного інтелекту та аналізу даних. Сьогодні ця мова активно використовується в медицині для обробки результатів діагностики та раннього виявлення захворювань, у фінансовому секторі – для прогнозування ринкових процесів і побудови автоматизованих торговельних систем, а також у маркетингу – для аналізу поведінки споживачів та створення персоналізованих рекомендацій. Універсальність і широкий спектр можливостей роблять Python одним із найефективніших інструментів для розроблення систем машинного навчання.

Бібліотеки

1. NumPy.

NumPy належить до базових бібліотек екосистеми Python для наукових обчислень та машинного навчання. Вона забезпечує роботу з багатовимірними масивами даних і надає широкий набір математичних функцій для виконання складних об-числень. Завдяки оптимізованій внутрішній структурі масиви NumPy працюють швидше та ефективніше за стандартні списки Python, що особливо важливо під час обробки великих наборів даних.

Бібліотека підтримує виконання численних операцій над масивами, включаючи зміну їх форми, транспонування, сортування та математичні перетворення. Це дозволяє суттєво спростити підготовку даних для подальшого аналізу та навчання моделей.

Seaborn є спеціалізованою бібліотекою для статистичної візуалізації даних у середовищі Python. Вона побудована на основі бібліотеки Matplotlib та має тісну інтеграцію з Pandas, що забезпечує зручну роботу з табличними даними.

Головною перевагою Seaborn є можливість швидкого створення інформативних графіків без необхідності детального налаштування параметрів візуалізації. Бібліотека автоматично виконує статистичну обробку даних та дозволяє будувати гістограми, діаграми розподілу, теплові карти, кореляційні матриці та інші графічні представлення результатів аналізу.

Для скорочення запису бібліотеку зазвичай імпортують під псевдонімом sns. Усі побудовані графіки використовують функціональні можливості Matplotlib, що забезпечує високу якість та гнучкість налаштувань.

2. Pandas.

Pandas є однією з найпопулярніших бібліотек для роботи зі структурованими даними. Вона надає інструменти для завантаження, очищення, аналізу та перетворення великих обсягів інформації.

Основними структурами даних у Pandas є Series та DataFrame, які дозволяють ефективно працювати з табличними даними. Бібліотека підтримує виконання опе-рацій групування, сортування, агрегації, об’єднання таблиць та обробки пропущених значень. Завдяки цьому Pandas значно спрощує процес підготовки даних перед використанням алгоритмів машинного навчання.

3. TensorFlow.

TensorFlow – це відкрита програмна платформа для створення та навчання моделей машинного й глибокого навчання. Вона містить широкий набір інструментів,

що дозволяють розробляти нейронні мережі різної складності та виконувати масштабні обчислення.

Однією з ключових особливостей TensorFlow є підтримка різних апаратних платформ, включаючи центральні процесори (CPU), графічні процесори (GPU) та тензорні процесори (TPU). Це забезпечує високу продуктивність під час навчання моделей на великих масивах даних.

Бібліотека широко застосовується для задач класифікації, прогнозування, розпізнавання образів, обробки природної мови та реалізації алгоритмів навчання з підкріпленням. Крім того, TensorFlow надає засоби для візуалізації процесу навчан-ня та оцінювання ефективності моделей [5].

4. Matplotlib.

Matplotlib є універсальною бібліотекою для побудови графіків і візуалізації результатів обробки даних. Вона дозволяє створювати як прості двовимірні графіки, так і складні інтерактивні візуалізації.

За допомогою Matplotlib можна будувати лінійні графіки, гістограми, кругові діаграми, теплові карти та інші типи графічних представлень. Бібліотека забезпечує широкі можливості налаштування зовнішнього вигляду графіків і часто використовується для аналізу даних перед етапом навчання моделей машинного навчання.

Важливою перевагою є сумісність із синтаксисом MATLAB, що полегшує перехід користувачів між цими середовищами.

5. Scikit-learn.

Scikit-learn є однією з найвідоміших бібліотек для реалізації алгоритмів класичного машинного навчання. Вона містить широкий набір методів навчання з учителем та без учителя, а також інструменти для попередньої обробки даних і оцінювання якості моделей.

Бібліотека підтримує алгоритми класифікації, регресії, кластеризації, зниження розмірності та відбору ознак. Однією з її переваг є тісна інтеграція з NumPy, Pandas та Matplotlib, що забезпечує зручність побудови повного циклу аналізу даних.

Scikit-learn активно використовується як у наукових дослідженнях, так і в промислових інформаційних системах. Завдяки простому інтерфейсу та якісній документації вона є популярним інструментом як серед досвідчених фахівців, так і серед початківців у галузі машинного навчання.

Проаналізувавши основні принципи проектування ІВС та можливості її інтеграції з алгоритмами машинного навчання, розробимо модель машинного навчання для прогнозування стану параметрів водного середовища.

Моделі побудована за допомогою бібліотека TensorFlow, данні для прогнозування взяті з відкритих джерел.

Спочатку підключаємо усі необхідні бібліотеки та завантажуємо данні для прогнозування:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from sklearn.model_selection import train_test_split
# load the data that is needed into the application
df = pd.read_csv('water_potability.csv')
df.head()
```

Виявляємо кореляцію (рис. 1).

```
sns.heatmap(df.corr(), cbar=True)
```

Після попереднього аналізу даних необхідно обробити пропущені значення, оскільки їх наявність може негативно впливати на результати статистичного аналізу та побудову моделей машинного навчання. Для цього використовується метод заповнення пропусків середніми значеннями відповідних стовпців. Такий підхід дозволяє зберегти всі записи в наборі даних та уникнути втрати інформації через видалення рядків із пропущеними значеннями.

Спочатку виконується заповнення пропусків у стовпці **ph** за допомогою команди:

```
df['ph'] = df['ph'].fillna(df['ph'].mean())
```

У цьому виразі:

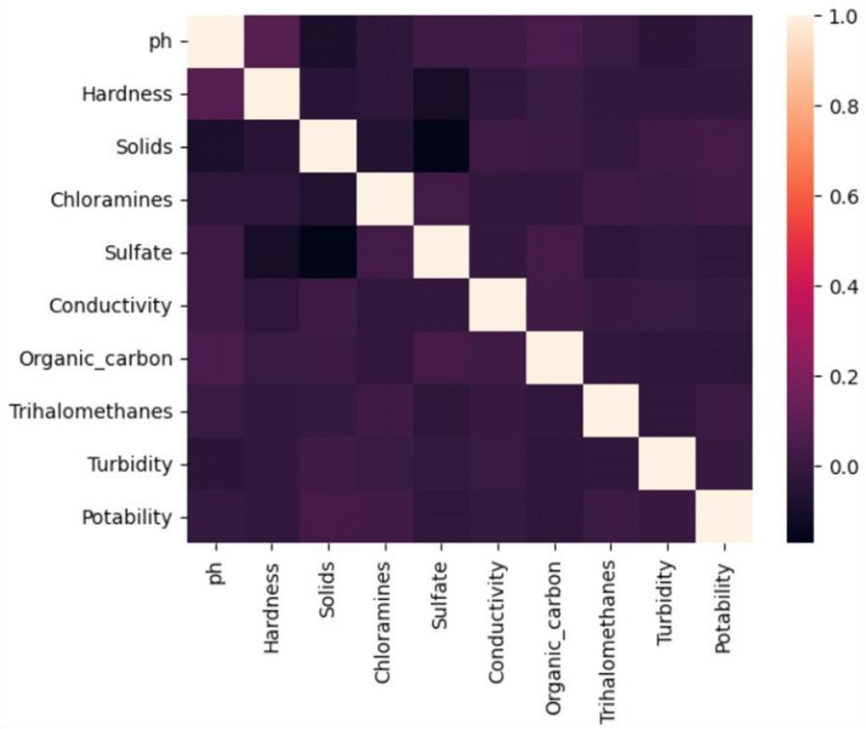

Рисунок 1 – Матриці кореляції

- df['ph'] – звернення до стовпця **ph** у DataFrame df;
- df['ph'].mean() – обчислення середнього арифметичного значення для всіх наявних елементів стовпця **ph**, при цьому пропущені значення (NaN) автоматично ігноруються;
- .fillna() – метод, який замінює всі відсутні значення на передане йому значення.

Таким чином, усі пропуски в стовпці **ph** замінюються середнім значенням цього показника, що дозволяє зберегти загальну структуру даних та мінімізувати вплив відсутніх значень на подальший аналіз.

Аналогічна процедура виконується для стовпця **Sulfate**:

```
df['Sulfate'] = df['Sulfate'].fillna(df['Sulfate'].mean())
```

Тут:

- df['Sulfate'] – вибірка стовпця **Sulfate** з набору даних;
- df['Sulfate'].mean() – розрахунок середнього значення концентрації сульфатів серед усіх доступних записів;
- .fillna() – заміна всіх пропущених значень на обчислене середнє.

У результаті всі відсутні значення в стовпці **Sulfate** замінюються середнім показником концентрації сульфатів, що забезпечує повноту даних для подальших розрахунків.

Далі аналогічний метод застосовується до стовпця **Trihalomethanes**:

```
df['Trihalomethanes'] = df['Trihalomethanes'].fillna(df['Trihalomethanes'].mean())
```

У цьому випадку:

- df['Trihalomethanes'] – вибірка стовпця **Trihalomethanes**;
- df['Trihalomethanes'].mean() – обчислення середнього значення вмісту тригалометанів;
- .fillna() – функція, що виконує заміну всіх значень NaN на розраховане середнє значення.

Після виконання цієї операції всі пропущені дані у стовпці **Trihalomethanes** будуть заповнені середнім значенням, характерним для цього показника.

Варто зазначити, що використаний підхід належить до найпоширеніших методів обробки пропущених даних. Його перевагою є простота реалізації та можливість

швидко підготувати набір даних до подальшого аналізу. Заповнення пропусків середніми значеннями дозволяє уникнути втрати спостережень і забезпечує коректну роботу алгоритмів машинного навчання, які зазвичай не можуть працювати з відсутніми значеннями. Після завершення цього етапу набір даних стає повним і готовим до проведення подальшого аналізу, візуалізації та побудови прогнозних моделей.

```
df.describe()
df.info()
df.isnull().sum()
missing_data = df.isnull().sum()
total = df.isnull().count()
percent = (missing_data/total) * 100
missing_data = pd.concat([missing_data, percent], axis=1, keys=['Total', 'Percent'])
missing_data
df['ph'] = df['ph'].fillna(df['ph'].mean())
df['Sulfate'] = df['Sulfate'].fillna(df['Sulfate'].mean())
df['Trihalomethanes'] = df['Trihalomethanes'].fillna(df['Trihalomethanes'].mean())
```

Обираємо прогнозований показник, у даному проекті був обраний показник придатності води до питного виживання так як він залежить від багатьох факторів що дає змогу на основі провести кореляцію між різними чинниками, а також цей показник є практичним для різних сфер застосування:

```
df.info()
X = df.drop('Potability', axis=1)
y = df['Potability']
```

Далі масштабуємо дані для пришвидшення навчання моделі, та розподіляємо їх:

```
# scale the data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X = scaler.fit_transform(X)
# split the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

train_test_split - функція яка розділяє дані на навчальну вибірку [6] (для тренування моделі) та тестову вибірку (для оцінки моделі).

Параметри: X - вхідні ознаки (фічі) після масштабування. Y - мітки (цільова змінна). test_size=0.2 - означає, що 20% даних буде виділено для тестової вибірки, а решта 80% — для навчання.

random_state=42 - фіксує початковий стан генератора випадкових чисел, щоб забезпечити відтворюваність результатів.

Команда розбиває масиви X та y на чотири частини: X_train - вхідні ознаки для навчальної вибірки (80% даних). X_test - вхідні ознаки для тестової вибірки (20% даних). y_train - мітки для навчальної вибірки. y_test - мітки для тестової вибірки. Тепер є два набори даних: навчальний набір - для тренування моделі та Тестовий набір для оцінки її продуктивності на нових, невідомих даних.

Далі будуємо модель машинного навчання використовуючи Tensorflow, та копіюємо її:

```
# build the model
model = tf.keras.models.Sequential([
tf.keras.layers.Dense(128, kernel_regularizer=tf.keras.regularizers.l2(0.001), activation='relu', input_dim=X_train.shape[1]),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(64, kernel_regularizer=tf.keras.regularizers.l2(0.001), activation= 'relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(1, activation= 'sigmoid')
])
# Compile the model
optimizer = tf.keras.optimizers.Adam(learning_rate=0.001)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
```

Побудована модель машинного навчання має багатошарову архітектуру нейронної мережі, яка складається з трьох основних шарів: вхідного, прихованого та вихідного. Така структура дозволяє ефективно виявляти складні залежності між

ознаками набору даних і виконувати бінарну класифікацію об'єктів.

Вхідний шар містить 128 нейронів і призначений для первинної обробки масштабованих вхідних даних. На цьому етапі кожен нейрон отримує інформацію про характеристики об'єктів дослідження та передає її для подальшого аналізу. Для підвищення якості навчання використовується функція активації ReLU (Rectified Linear Unit), яка забезпечує нелінійність моделі та дозволяє нейронній мережі виявляти складні закономірності в даних. Додатково застосовується L2-регуляризація, що обмежує надмірне зростання вагових коефіцієнтів і сприяє кращій узагальнювальній здатності моделі.

Прихований шар складається з 64 нейронів і виконує поглиблену обробку отриманої інформації. Порівняно з попереднім шаром кількість нейронів зменшується, що дозволяє сформувати більш компактне та інформативне представлення даних. Для запобігання перенавчанню використовується метод Dropout, який під час кожної ітерації випадковим чином вимикає частину нейронів із процесу навчання. Це змушує мережу шукати більш стійкі закономірності та підвищує її здатність працювати з новими даними. Також у прихованому шарі застосовується L2-регуляризація, яка додатково контролює складність моделі та знижує ризик запам'ятовування тренувальних даних.

Вихідний шар містить один нейрон, оскільки модель розв'язує задачу бінарної класифікації. Для цього використовується функція активації Sigmoid, яка перетворює результат роботи мережі на значення в діапазоні від 0 до 1. Отримане значення інтерпретується як ймовірність належності об'єкта до позитивного класу. На основі цієї ймовірності формується остаточне рішення щодо класифікації об'єкта.

Використання комбінації L2-регуляризації та Dropout є ефективним засобом боротьби з перенавчанням, оскільки дозволяє зменшити залежність моделі від особливостей тренувальної вибірки та покращити її здатність до узагальнення. Водночас функція активації ReLU сприяє швидкому та стабільному навчанню мережі, а Sigmoid забезпечує коректне формування ймовірнісного прогнозу для задач бінарної класифікації. Завдяки такій архітектурі модель здатна ефективно аналізувати вхідні дані та формувати точні прогнози щодо належності об'єктів до відповідних класів.

Налаштовуємо та тренуємо модель:

```
# callbacks
early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
# Train the model
history = model.fit(X_train, y_train, epochs=200, batch_size=32, validation_data=(X_test, y_test), callbacks=[early_stopping])
# Model Evaluation
evaluation = model.evaluate(X_test, y_test)
print(f"Test Loss: {evaluation[0]}, Test Accuracy: {evaluation[1]}")
# Summary of the process and results
import matplotlib.pyplot as plt
```

Після підготовки даних та побудови архітектури нейронної мережі виконується етап навчання моделі. Для цього використовується метод fit(), який запускає процес підбору вагових коефіцієнтів на основі тренувальної вибірки. Основними параметрами навчання є вхідні дані, кількість епох, розмір пакета даних, валідаційна вибірка та механізм ранньої зупинки навчання.

Параметр X_train містить набір вхідних даних для тренування моделі. Це незалежні змінні або ознаки, які характеризують кожен об'єкт дослідження та використовуються алгоритмом для пошуку закономірностей. У задачі прогнозування якості води ці ознаки можуть включати значення рівня pH, жорсткості води, концентрації сульфатів, тригалометанів та інших показників.

Параметр y_train містить цільові значення (мітки класів), які відповідають кожному запису в наборі X_train. Саме ці значення модель намагається передбачити під час навчання. У даному дослідженні вони представляють залежну змінну, яка визначає належність зразка води до певного класу якості.

Для навчання моделі встановлено параметр:

```
epochs=200
```

Параметр epochs визначає кількість повних проходів моделі через увесь тренувальний набір даних. Одна епоха означає, що кожен запис із тренувальної вибірки був використаний один раз для оновлення ваг нейронної мережі. У даному випадку максимальна кількість епох становить 200. Проте фактична кількість ітерацій може бути меншою завдяки використанню механізму ранньої зупинки (Early Stopping), який автоматично припиняє навчання, якщо якість моделі перестає покращуватися.

Наступним важливим параметром є:

```
batch_size=32
```

Параметр batch_size визначає кількість прикладів, які одночасно подаються до нейронної мережі для одного кроку оновлення ваг. У цьому випадку весь набір тренувальних даних розбивається на невеликі групи по 32 записи. Після обробки кожного такого пакета виконується коригування вагових коефіцієнтів моделі. Використання пакетів невеликого розміру дозволяє ефективніше використовувати обчислювальні ресурси та часто сприяє кращій узагальнювальній здатності моделі. Водночас занадто малий розмір пакета може збільшувати коливання процесу навчання, тому значення 32 є одним із найпоширеніших компромісних варіантів.

Для контролю якості навчання використовується параметр:

```
validation_data=(X_test, y_test)
```

Тут X_test містить ознаки тестової вибірки, які не використовуються для навчання моделі. Їхнє призначення полягає у перевірці здатності моделі працювати з новими даними, яких вона раніше не бачила. Відповідно, y_test містить правильні відповіді для кожного об’єкта тестової вибірки.

Після завершення кожної епохи модель виконує оцінювання своєї продуктивності на тестових даних. Це дозволяє відстежувати зміну значень функції втрат та точності не лише на тренувальних даних, а й на незалежній вибірці. Такий підхід дає можливість виявити ознаки перенавчання, коли модель демонструє хороші результати на тренувальних даних, але погано працює на нових прикладах.

Для автоматичного контролю процесу навчання використовується список зворотних викликів:

```
callbacks=[early_stopping]
```

У даному випадку список містить механізм Early Stopping, який здійснює моніторинг якості моделі на валідаційних даних. Якщо протягом визначеної кількості епох значення контрольного показника не покращується, процес навчання автоматично припиняється. Це дозволяє уникнути перенавчання моделі, скоротити час навчання та зберегти найкращу версію мережі.

Після завершення процесу навчання проводиться аналіз історії зміни функції втрат (loss) для тренувальної та валідаційної вибірок. На основі отриманих даних будується графік порівняння втрат навчання та перевірки (рис. 2). Аналіз такого графіка дозволяє оцінити стабільність процесу навчання, визначити момент досягнення оптимальної якості моделі та перевірити відсутність перенавчання. Якщо криві тренувальних і валідаційних втрат мають схожу динаміку та не демонструють значного розходження, це свідчить про хорошу здатність моделі до узагальнення та коректну роботу алгоритму на нових даних.

```
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(['Train', 'Validation'], loc='upper left')
```

Далі проведемо прогнозування за дпомогою навченої моделі та проаналізуємо ефективність роботи моделі за допомогою матриці плутання (рис. 3):

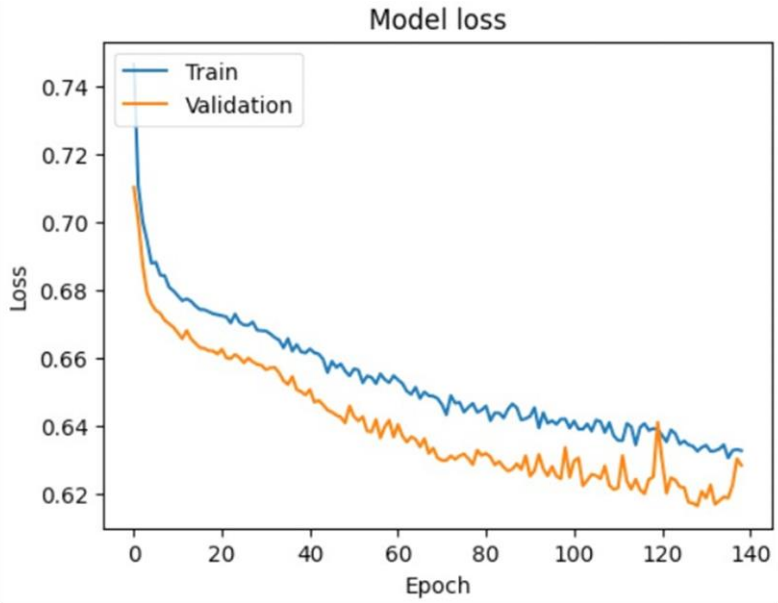

Рисунок 2 – Графік порівняння втрати перевірки та навчання

```
predictions = model.predict(X_test)
predictions = np.round(predictions)
from sklearn.metrics import class-
?fication_report, confusion_matrix
print(classification_report(y_test, predi-
?tions))
print(confusion_matrix(y_test, predictions))
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, predi-
?tions)
print(f"Accuracy: {accuracy}")
from sklearn.metrics import precision_score
precision = precision_score(y_test,
predictions)
print(f"Precision: {precision}")
cm = confusion_matrix(y_test, predictions)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap='Blues',
fmt='g')
plt.xlabel('Predictions')
plt.ylabel('Actuals')
plt.title('Confusion Matrix')
plt.show()
```

predictions - отримує прогнозовані значення для тестових даних (X_test).

Після завершення навчання нейронної мережі виконується оцінювання її ефективності на тестовій вибірці. Для цього спочатку отримуються прогнозовані значення за допомогою методу predict(), який повертає ймовірності належності кожного об’єкта до певного класу.

Результатом виконання прогнозування є масив числових значень у діапазоні від 0 до 1. Це пояснюється тим, що у вихідному шарі нейронної мережі використовується функція активації sigmoid, яка перетворює результат роботи моделі на ймовірність належності об’єкта до позитивного класу. Чим ближче отримане значення до 1, тим вища впевненість моделі у тому, що об’єкт належить до позитивного класу. Відповідно, значення, близькі до 0, свідчать про належність до негативного класу.

Для перетворення отриманих ймовірностей у конкретні класи використовується команда:

```
np.round(predictions)
```

Функція np.round() виконує округлення прогнозованих значень до найближчого цілого числа. У задачах бінарної класифікації це дозволяє отримати остаточні результати прогнозування у вигляді класів 0 та 1. Якщо значення прогнозу є більшим або дорівнює 0,5, воно округлюється до 1, що відповідає позитивному класу. Якщо значення є меншим за 0,5, результат округлюється до 0, тобто модель відносить об’єкт до негативного класу. Таким чином, імовірнісні прогнози перетворюються на зрозумілі класи, які можна порівнювати з реальними значеннями.

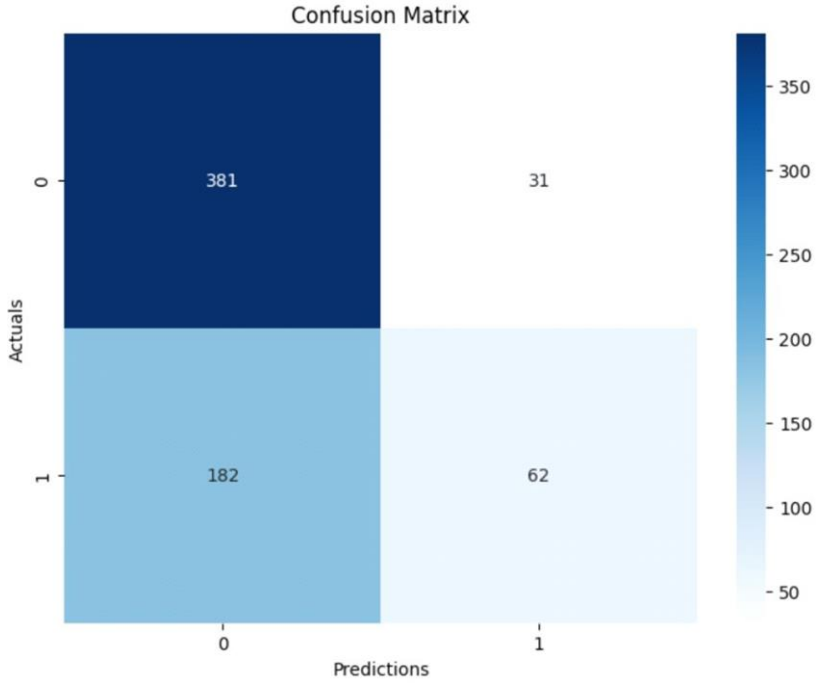

Рисунок 3 – Матриця плутань

Для комплексної оцінки якості класифікації використовується функція classification_report(), яка формує детальний звіт щодо основних метрик моделі. Цей звіт містить такі показники:

Precision (точність) – характеризує частку правильно визначених позитивних об'єктів серед усіх об'єктів, які модель віднесла до позитивного класу. Високе значення точності свідчить про малу кількість хибнопозитивних прогнозів.

Recall (повнота або відгук) – показує частку правильно виявлених позитивних об'єктів серед усіх реально позитивних прикладів. Ця метрика є важливою у випадках, коли необхідно мінімізувати кількість пропущених позитивних результатів.

F1-score – являє собою гармонійне середнє між показниками точності та повноти. Дана метрика дозволяє оцінити баланс між цими двома характеристиками та є особливо корисною при роботі з незбалансованими даними.

Support – відображає кількість реальних спостережень для кожного класу в тестовій вибірці. Цей показник дозволяє оцінити репрезентативність даних для кожного класу.

Для більш детального аналізу результатів класифікації використовується функція confusion_matrix(), яка формує матрицю сплутаних результатів (матрицю помилок). Вона демонструє кількість правильних і неправильних прогнозів для кожного класу та дозволяє оцінити характер помилок моделі.

Матриця сплутаних результатів складається з чотирьох основних елементів:

True Positive (TP) – кількість позитивних об'єктів, які модель правильно класифікувала як позитивні;

True Negative (TN) – кількість негативних об'єктів, які модель правильно класифікувала як негативні;

False Positive (FP) – кількість негативних об'єктів, які модель помилково віднесла до позитивного класу;

False Negative (FN) – кількість позитивних об'єктів, які модель помилково віднесла до негативного класу.

На основі цих значень можна більш детально проаналізувати сильні та слабкі сторони моделі та визначити напрямки для її подальшого вдосконалення.

Для оцінки загальної ефективності алгоритму використовується метрика accuracy_score(), яка обчислює загальну точність класифікації. Вона визначається як відношення кількості правильно класифікованих об'єктів до загальної кількості

записів у тестовій вибірці. Значення цієї метрики показує, який відсоток усіх прогнозів був виконаний правильно. Чим вищий показник accuracy, тим кращою є загальна якість моделі.

Додатково розраховується показник precision_score(), який оцінює точність позитивних прогнозів. Ця метрика визначає, яка частка об'єктів, віднесених моделлю до позитивного класу, дійсно є позитивними. Високе значення precision свідчить про те, що модель рідко помиляється під час визначення позитивного класу та формує надійні прогнози.

Отримані значення метрик дозволяють комплексно оцінити ефективність побудованої нейронної мережі, визначити якість її прогнозів та зробити висновки щодо можливості практичного використання моделі для автоматизованого прогнозування якості води.

Проаналізувавши отриману матрицю ми може констатувати що модель працює коректно, так як кількість правильно спрогнозованих значень перевищує кількість невірно спрогнозованих значень.

Висновки

У ході виконання роботи було проаналізовано сучасні підходи до побудови та використання інформаційно-вимірювальних систем (ІВС), а також досліджено можливості їх удосконалення шляхом інтеграції технологій машинного навчання. Проведений аналіз наукової літератури дозволив визначити основні тенденції розвитку інтелектуальних систем моніторингу та виявити актуальні проблеми контролю якості водних ресурсів. Особливу увагу було приділено питанням забруднення водного середовища, оскільки своєчасне виявлення відхилень показників якості води є важливим завданням для забезпечення екологічної безпеки та захисту здоров'я населення.

На основі проведеного дослідження встановлено, що інтеграція методів машинного навчання в інформаційно-вимірювальні системи є перспективним напрямом розвитку сучасних систем моніторингу. Використання алгоритмів штучного інтелекту дозволяє автоматизувати процес аналізу великих обсягів даних, підвищити точність прогнозування та оперативність прийняття рішень. Крім того, такі системи здатні виявляти приховані закономірності в даних і прогнозувати можливі зміни стану контрольованого середовища ще до виникнення критичних ситуацій.

У рамках практичної частини роботи було розроблено та протестовано модель машинного навчання на базі бібліотеки TensorFlow для прогнозування стану параметрів водного середовища. Виконано підготовку та обробку даних, проведено навчання нейронної мережі, а також оцінено її ефективність за допомогою основних метрик класифікації. Отримані результати підтвердили працездатність запропонованого підходу та продемонстрували можливість використання нейронних мереж для розв'язання задач моніторингу якості води. Практична цінність роботи полягає в тому, що розроблену модель можна використовувати як основу для створення інтелектуальних інформаційно-вимірювальних систем екологічного моніторингу. Подальший розвиток дослідження може бути спрямований на використання більших обсягів даних, удосконалення архітектури нейронної мережі, інтеграцію моделі з реальними сенсорними системами збору даних та розробку комплексних програмно-апаратних рішень для автоматизованого контролю стану водних ресурсів у режимі реального часу.

Таким чином, поставлені в роботі завдання були виконані, а отримані результати підтверджують доцільність використання технологій машинного навчання для підвищення ефективності функціонування сучасних інформаційно-вимірювальних систем контролю якості водного середовища.

Подяки

Відсутні.

Конфлікт інтересів

Відсутній.

Список використаних джерел

1. Шестопалов О. В., Сакун А. О., Лізантан П. С., Кануннікова Н. О., Гайдучек О. Г., Томашевський Р. С., Воробйов Б. В. Аналіз показників якості води: сучасні аспекти та виклики. *Екологічні науки*. 2024. № 3. С. 76–82. URL: https://doi.org/10.32846/2306-9716/2024.eco.3-54.10

2. Погребенник В. Д., Романюк А. В. Інформаційно-вимірювальна система для оперативного екологічного моніторингу водного середовища. *Вимірювальна техніка та метрологія*. 2009. № 70. URL: https://vlp.com.ua/files/08_58.pdf

3. Мокієнко А. В., Бабієнко В. В., Гущук І. В. Клімат, вода та інфекції: нові виклики для півдня України на тлі старих проблем. *Public Health Journal*. 2023. № 4. URL: https://doi.org/10.32782/pub.health.2023.4.6

4. Гіроль М. М., Ковальський Д., Хомко В. Є., Гіроль А. М. Проблеми якості води в водопровідних мережах. *Водопостачання та водовідведення*. 2008. № 2. С. 1–21.

5. Prokopov O., Lypovetska O., Kulish T. Dangerous chlorites in drinking water: formation and removal using chlorine dioxide in water treatment technology. *Environment and Health*. 2023. № 1.

6. Документація мови Python. URL: https://docs.python.org/3/

7. Документація TensorFlow. URL: https://www.tensorflow.org/api_docs

8. Документація Scikit-Learn. URL: https://scikit-learn.org/stable/

9. Breiman L. Random forests. *Machine Learning*. 2001. Vol. 45. P. 5–32. URL: https://doi.org/10.1023/A:1010933404324

10. Nilsson N. J. Introduction to machine learning. 2005. URL: https://pzs.dstu.dp.ua/DataMining/bibl/MLBOOK.pdf

11. Дорошенко А. Ю., Шпиг В. М., Кушніренко Р. В. Застосування машинного навчання для уточнення чисельних метеорологічних прогнозів. 2020. URL: https://ceur-ws.org/Vol-2866/ceur_375-383doroshenko_shpig38.pdf

References

1. Shestopalov, O. V., Sakun, A. O., Lizantan, P. S., Kanunnikova, N. O., Haiduchek, O. G., Tomashevskyi, R. S., & Vorobiov, B. V. (2024). Analiz pokaznykiv yakosti vody: suchasni asekty ta vyklyky [Analysis of water quality indicators: Modern aspects and challenges]. *Ekolohichni Nauky*, (3), 76–82. https://doi.org/10.32846/2306-9716/2024.eco.3-54.10 (in Ukrainian)

2. Pohrebendnyk, V. D., & Romaniuk, A. V. (2009). Informatsiino-vymiriuvalna systema dlia operatyvnoho ekolohichnoho monitorynhu vodnoho seredovyshcha [Information and measurement system for online environmental monitoring of water environment]. *Vymiriuvalna Tekhnika ta Metrolohiia*, (70), 54–58. https://vlp.com.ua/files/08_58.pdf (in Ukrainian)

3. Mokienko, A. V., Babiienko, V. V., & Hushchuk, I. V. (2023). Klimat, voda ta infektsii: novi vyklyky dlia pivdnia Ukrainy na tli starykh problem [Climate, water and infections: New challenges for the south of Ukraine against the background of old problems]. *Public Health Journal*, (4), 48–56. https://doi.org/10.32782/pub.health.2023.4.6 (in Ukrainian)

4. Hirol, M. M., Kovalskyi, D., Khomko, V. Ye., & Hirol, A. M. (2008). Problemy yakosti vody v vodoprovidnykh merezhakh [Water quality problems in water supply networks]. *Vodopostachannia ta Vodovidvedennia*, (2), 1–21. (in Ukrainian)

5. Prokopov, O., Lypovetska, O., & Kulish, T. (2023). Dangerous chlorites in drinking water: Formation and removal using chlorine dioxide in water treatment technology. *Environment and Health*, (1), 40–47.

6. Python Software Foundation. (n.d.). *Python language documentation*. https://docs.python.org/3/

7. Google. (n.d.). *TensorFlow documentation*. https://www.tensorflow.org/api_docs
8. Scikit-learn Developers. (n.d.). *Scikit-learn documentation*. https://scikit-learn.org/stable/
9. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324
10. Nilsson, N. J. (2005). *Introduction to machine learning*. Stanford University. https://pzs.dstu.dp.ua/DataMining/bibl/MLBOOK.pdf
11. Doroshenko, A. Yu., Shpyh, V. M., & Kushnirenko, R. V. (2020). Zastosuvannia mashynnoho navchannia dlia utochnennia chyslennykh meteorolohichnykh prohnoziv [Application of machine learning for refinement of numerical meteorological forecasts]. *CEUR Workshop Proceedings*, *2866*, 375–383. https://ceur-ws.org/Vol-2866/ceur_375-383doroshenko_shpig38.pdf (in Ukrainian)

INFORMATION AND MEASURING SYSTEM OF OPERATIONAL WATER QUALITY BASED ON MACHINE LEARNING

Demchyna M. M.
Candidate of Technical Sciences, Associate Professor
King Danylo University
76018, Konovaltsya St., 35, Ivano-Frankivsk, Ukraine
https://orcid.org/0009-0002-9161-4843
e-mail: mykolademchyna@gmail.com

Abstract. Over the past few decades, there have been significant improvements in the methods of numerical modeling used to predict environmental processes in natural systems. This progress has been made possible by the rapid advancement of computing technology, increased accuracy of mathematical models, and the active integration of modern data analysis methods into scientific research. Today, machine learning technologies play a key role among environmental monitoring tools, providing new opportunities for processing and interpreting large datasets on the state of the environment. The application of intelligent algorithms makes it possible not only to automate the process of data collection and analysis but also to identify hidden relationships between environmental indicators. This facilitates a deeper investigation of complex natural systems characterized by high dynamism and multifactoriality. Of particular importance is the ability to forecast climate change, as well as to analyze hydrometeorological and biometric parameters, which allows for an objective assessment of the state of the environment and the formulation of scientifically sound management decisions. The combination of modern hydrodynamic models with intelligent data processing methods helps improve the reliability of forecasts and reduce uncertainty when modeling environmental processes. Satellite observation systems, automated monitoring stations, and sensor networks play a key role in this, as they provide comprehensive data on water circulation, pollutant migration, temperature changes, and other processes that determine the ecological condition of water bodies. Thus, the integration of traditional mathematical modeling approaches with modern data analysis technologies creates an effective foundation for the development of high-precision environmental monitoring systems. Further development in this area, an increase in the volume of available information, and the expansion of interdisciplinary cooperation will contribute to a deeper understanding of natural processes and improve the effectiveness of environmental protection activities at the local, regional, and national levels.

Keywords: machine learning, aquatic environment, ecology, forecasting, information and measurement system.