МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ГОНЧАР Ярослав Андрійович

Методи відновлення відсутніх даних на основі нейронних мереж / Methods of Missing Data Recovery Based on Neural Networks

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КНм-21
Я.А. Гончар
_____

Науковий керівник:
к.т.н., доцент Х.В. Ліп'яніна-Гончаренко
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
В.о. завідувача кафедри
_____ Н.М. Васильків

ТЕРНОПІЛЬ - 2024

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Гончар Ярослав Андрійович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Методи відновлення відсутніх даних на основі нейронних мереж / Methods of Missing Data Recovery Based on Neural Networks
керівник роботи к.т.н., доцент Х.В. Ліп'яніна-Гончаренко
затверджені наказом по університету від 12 грудня 2023 року № 753.
2. Строк подання студентом закінченої кваліфікаційної роботи 4 грудня 2024 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити
- опис предметної області;
- аналіз відомих методів обробки пропущених даних;
- аналіз метрик оцінки ефективності моделей машинного навчання;
- постановка задачі дослідження;
- підхід до відновлення відсутніх даних на основі алгоритму нечіткої кластеризації та згорткової нейронної мережі;
- архітектура згорткової нейронної мережі;
- постановка експериментальних досліджень;
- набори даних та налаштування параметрів;
- критерії оцінки ефективності;
- результати експериментальних досліджень.

5. Перелік графічного матеріалу у роботі
- схема роботи підходу до імпутації даних;
- схема процесу імпутації пропущених значень;
- архітектура згорткової нейронної мережі для імпутації пропущених даних.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 12 грудня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи.	до 01.01. 2024 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.03. 2024 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 20.05.2024 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 28.10. 2024 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 11.11.2024 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів.	до 25.11.2024 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту.	до 30.11.2024 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 04.12.2024 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії.	до 14.12. 2024 р.	

Студент _____ Я.А. Гончар
підпис

Керівник роботи _____ к.т.н., доцент Х.В. Ліп’яніна-Гончаренко
підпис

РЕЗЮМЕ

Кваліфікаційна робота на тему «Методи відновлення відсутніх даних на основі нейронних мереж» на здобуття освітнього ступеня «Магістр» зі спеціальності 122 «Комп’ютерні науки» освітньої програми «Комп’ютерні науки» написана обсягом в 74 сторінки і містить 10 ілюстрацій, 3 таблиці, 3 додатки та 54 використаних джерел.

Метою кваліфікаційної роботи є розробка ефективного підходу до імпутації пропущених даних на основі згорткових нейронних мереж, який враховує просторові та нелінійні зв’язки між ознаками

Методи досліджень: методи аналізу та синтезу, методи математичного моделювання, кластеризації, експериментальні методи.

Результати дослідження: вдосконаленно метод імпутації пропущених даних шляхом застосування згорткових нейронних мереж, що враховують просторові та нелінійні зв’язки між ознаками у наборі даних, що дозволяє покращити точність відновлення пропущених значень, забезпечуючи більш ефективне навчання моделей класифікації навіть при значних обсягах пропусків.

Результати роботи можуть успішно застосовуватися для покращення якості даних і підвищення точності моделей класифікації в реальних умовах, де часто трапляються пропуски даних. Метод може бути інтегрований у системи, що працюють з великими даними, для автоматизації процесів імпутації та зменшення похибок у прогнозуванні.

Ключові слова: ВІДСУТНІ ДАНІ, ІМПУТАЦІЯ ПРОПУЩЕНИХ ДАНИХ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, НАБІР ДАНИХ, ЯКІСТЬ ДАНИХ.

ABSTRACT

Qualification work on the topic «Methods of Missing Data Recovery Based on Neural Networks» for Master's degree on speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 74 pages and it contains 10 figures, 3 tables, 3 annexes and 54 sources.

The purpose of this qualification work is to develop an effective approach to imputing missing data based on convolutional neural networks, which takes into account spatial and nonlinear relationships between features.

Research methods: methods of analysis and synthesis, mathematical modeling, clustering, and experimental methods.

Research results: The method of imputing missing data has been improved through the application of convolutional neural networks, which consider spatial and nonlinear relationships between features in the dataset. This approach enhances the accuracy of restoring missing values, enabling more effective training of classification models, even in cases of significant missing data volumes.

The results of the work can be successfully applied to improve data quality and increase the accuracy of classification models in real-world scenarios where missing data frequently occur. The method can be integrated into systems working with big data to automate imputation processes and reduce errors in predictions.

Keywords: MISSING DATA, MISSING DATA IMPUTATION, CONVOLUTIONAL NEURAL NETWORK, DATASET, DATA QUALITY.

ЗМІСТ

ВСТУП

Актуальність роботи. У наш час у багатьох сферах поступово накопичується велика кількість даних. Це стрімке зростання даних підкреслює важливість їх аналізу, особливо коли йдеться про великі обсяги інформації. Проте важливо, щоб зібрані дані були надійними та корисними, оскільки дані поганої якості не можуть бути основою для побудови точних моделей.

Одна з поширених проблем у роботі з даними – це наявність пропущених значень, що може створити проблеми під час аналізу. Такі пропущені дані трапляються в різних сферах, таких як аналіз експресії генів, контроль дорожнього руху, промислові інформаційні системи, обробка зображень і розробка програмного забезпечення. Якщо цю проблему не врахувати під час аналізу, це може призвести до неправильних висновків і результатів. Тому важливо підвищити якість даних, обробляючи належним чином пропущені значення.

Існують два основні традиційні методи обробки пропущених даних. Перший метод полягає в тому, щоб просто видалити всі записи, де відсутні якісь дані. Другий метод полягає в тому, щоб замінити пропущені значення на якісь припустимі значення. Крім того, існують методи імпутації (заміщення) даних, які використовують машинне навчання. Наприклад, до таких методів належать метод найближчих сусідів (KNN), рекурентні нейронні мережі (RNN), і генеративні змагальні мережі (GAN) для імпутації даних.

За останні кілька років глибоке навчання стало широко використовуватися для вирішення різних задач, включаючи заміщення пропущених даних. Використання великих обсягів навчальних даних дозволило значно покращити результати імпутації. Зокрема, GAN показали великі успіхи у вирішенні цієї задачі. Наприклад, один із методів імпутації на основі GAN вимагав налаштування гіперпараметрів для регулювання функції втрат і швидкості роботи дискримінатора. В іншому підході GAN використовував генератор і дискримінатор окремо для навчання структури і розподілу пропущених даних. Хоча ці методи дають чудові результати, вони часто занадто складні для

практичного використання через велику кількість налаштувань. Тому дослідження в даному напрямку є актуальними.

Мета і завдання дослідження. Метою кваліфікаційної роботи є розробка ефективного підходу до імпутації пропущених даних на основі згорткових нейронних мереж, який враховує просторові та нелінійні зв’язки між ознаками. Запропонований метод має на меті покращити якість даних і підвищити точність класифікаційних моделей, забезпечуючи стабільну продуктивність навіть за умов значної кількості пропусків у великих наборах даних.

Завдання кваліфікаційної роботи:

1. Провести аналіз сучасних підходів до імпутації пропущених даних з метою визначення їх переваг і недоліків у контексті відновлення пропущених значень та точності класифікації.

2. Розробити метод імпутації на основі згорткової нейронної мережі, яка враховує просторові і нелінійні зв’язки у даних для точнішого заповнення пропущених значень.

3. Використати кластеризацію для покращення організації даних та підвищення ефективності навчання моделі.

4. Дослідити, як врахування просторових відносин між ознаками у матриці даних впливає на якість імпутації. Особлива увага приділяється тому, як такі зв’язки можуть бути використані для покращення результатів прогнозування.

5. Дослідити стабільність запропонованого підходу при різних рівнях пропусків (від 10% до 70%) у даних і перевірити, чи впливає зростання кількості пропусків на точність імпутації та подальшої класифікації.

6. Провести серію експериментів оцінки ефективності запропонованого методу у порівнянні з іншими методами імпутації. Оцінити точність класифікації, отриману після імпутації пропущених даних за допомогою різних алгоритмів навчання з учителем.

7. Провести аналіз різних конфігурацій архітектури згорткових нейронних мереж, таких як кількість шарів, розмір фільтра, кількість кластерів, та їхній вплив на точність імпутації і класифікації. Визначити оптимальні налаштування для досягнення найкращих результатів.

Об'єктом дослідження є процес імпутації пропущених даних у наборах даних для підвищення ефективності машинного навчання, зокрема у контексті використання згорткових нейронних мереж для відновлення та покращення якості даних.

Предметом дослідження є методи та алгоритми імпутації пропущених даних, зокрема на основі згорткових нейронних мереж, що враховують просторові та нелінійні зв’язки для покращення точності класифікаційних моделей у наборах даних з високим рівнем пропусків.

Методи досліджень: методи аналізу та синтезу для огляду сучасних підходів до імпутації пропущених даних, методи математичного моделювання для розробки алгоритму на основі згорткових нейронних мереж, кластеризація для покращення організації даних, а також експериментальні методи для оцінки ефективності запропонованого підходу в порівнянні з іншими методами імпутації на різних наборах даних.

Наукова новизна одержаних результатів полягає у вдосконаленні методу імпутації пропущених даних шляхом застосування згорткових нейронних мереж, що враховують просторові та нелінійні зв’язки між ознаками у наборі даних, що дозволяє покращити точність відновлення пропущених значень, забезпечуючи більш ефективне навчання моделей класифікації навіть при значних обсягах пропусків.

Практичне значення отриманих результатів полягає у можливості застосування розробленого методу імпутації пропущених даних на основі згорткових нейронних мереж для обробки великих обсягів даних у різних галузях, де важлива точність відновлення пропущених значень. Запропонований підхід може бути використаний для покращення якості даних і підвищення точності моделей класифікації в реальних умовах, де часто трапляються пропуски даних. Крім того, метод може бути інтегрований у системи, що працюють з великими даними, для автоматизації процесів імпутації та зменшення похибок у прогнозуванні.

Публікації та апробація КР. Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах:

– міжнародної мультидисциплінарної наукової інтернет-конференції «Світ наукових досліджень» (випуск 35), 20-21 листопада 2024 р.;

– міжнародної наукової Інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення» (випуск 94), 11-12 грудня 2024 р.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПІДХОДІВ ДО ОБРОБКИ ПРОПУЩЕНИХ ДАНИХ

1.1 Опис предметної області

Предметна область, пов'язана з обробкою пропущених даних та методами їх імпутації, охоплює широкий спектр тем у галузі аналізу даних, машинного навчання, математичної статистики, глибинного навчання та штучного інтелекту. В сучасних умовах кількість даних, з якими працюють аналітики, науковці та розробники систем штучного інтелекту, швидко зростає, і забезпечення якості цих даних стає одним із ключових викликів. Пропущені дані є серйозною проблемою у багатьох сферах, починаючи від бізнес-аналітики до медицини, фінансів та прогнозування кліматичних умов. Відсутність даних може суттєво вплинути на точність прогнозів, прийняття рішень та ефективність систем машинного навчання.

Пропущені дані є поширеним явищем у будь-якій реальній базі даних або інформаційній системі. Вони виникають через різні причини, такі як людські помилки, технічні збої, неспроможність отримати певну інформацію або свідоме рішення не включати певні дані. Пропуски можуть бути випадковими (MCAR - Missing Completely at Random), залежати від спостережуваних даних (MAR - Missing at Random) або залежати від неспостережуваних чинників (MNAR - Missing Not at Random). Від того, як виникли пропуски, залежить ефективність різних методів їх імпутації.

Пропуски у даних можуть спричиняти значні проблеми під час аналізу даних і побудови моделей машинного навчання. Якщо пропущені значення не обробити належним чином, це може призвести до таких проблем:

1. Зниження якості прогнозів і результатів: відсутність частини даних може призвести до втрати важливих залежностей між ознаками та мітками, що негативно вплине на прогнози моделей.

2. Викривлення статистичних висновків: без коректного заповнення пропущених даних статистичні оцінки, такі як середні, медіани або частоти, можуть бути спотвореними.

3. Зменшення ефективності навчання моделей: багато алгоритмів машинного навчання не можуть працювати з неповними наборами даних, що змушує дослідників або відкидати записи з пропущеними значеннями, або використовувати методи імпутації.

Сучасні методи імпутації можна поділити на кілька категорій:

1. Статистичні методи:

1.1. Заміщення середнім. Це один із найпростіших методів, де пропущені значення заповнюються середнім для ознаки. Цей метод легкий в реалізації, але його основний недолік полягає у втраті варіаційності в даних і викривленні розподілів ознак.

1.2. Заміщення модою або медіаною. У випадках з категоріальними змінними використовують заміщення модою, а для числових даних, що мають перекошений розподіл – медіаною.

1.3. Регресійна імпутація. Метод полягає у використанні інших ознак для прогнозування пропущених значень через побудову регресійної моделі.

2. Методи машинного навчання [8, 13, 39]:

2.1. KNN. Цей метод заповнює пропуски, знаходячи найближчих сусідів у просторі ознак і використовує їхні значення для прогнозування пропущених даних. Однак, метод KNN може бути обчислювально дорогим для великих наборів даних і потребує коректного визначення метрики для вимірювання відстані між записами.

2.2. Метод випадкових лісів (RandomForest). Цей підхід будує ансамбль рішень (дерев) для прогнозування пропущених значень. Випадкові ліси можуть враховувати взаємозв'язки між багатьма ознаками і генерують більш точні прогнози, ніж простіші методи.

2.3. Множинна імпутація за допомогою ланцюгових рівнянь (MICE). Цей метод створює кілька наборів імпутованих даних і застосовує до них різні алгоритми для зменшення невизначеності при заповненні пропусків.

3. Глибоке навчання та нейронні мережі [15, 27]:

3.1. Автоенкодери. Архітектура автоенкодера використовується для навчання моделей на неповних даних шляхом мінімізації похибки реконструкції.

Модель відтворює повний набір даних, в тому числі заповнюючи пропущені значення, зменшуючи похибки між початковими та імпутованими даними [5, 15, 24, 27, 36, 47].

3.2. Згорткові нейронні мережі (CNN). Згорткові нейронні мережі використовуються для обробки даних з просторовими та нелінійними зв'язками між ознаками. Вони дозволяють ефективно імпутувати пропущені дані, використовуючи просторову інформацію та кореляції між ознаками.

3.3. GAN. Вони використовують два блоки (генератор і дискримінатор) для навчання моделі, яка може генерувати пропущені дані таким чином, щоб вони виглядали схожими на справжні дані.

Основними викликами є ефективна обробка пропущених даних у великих наборах даних, де існують складні взаємозв'язки між ознаками. Методи, які враховують просторові кореляції, такі як згорткові нейронні мережі, мають значний потенціал для підвищення точності імпутації. Однак ці методи вимагають значних обчислювальних ресурсів і можуть бути складними в налаштуванні.

Додатковий виклик полягає в забезпеченні стійкості методів імпутації при великій кількості пропусків у даних (наприклад, понад 50%). Для цього необхідно розробляти стратегії, що враховують структуру даних, типи змінних, а також можливість змішування різних методів, таких як комбінування автоенкодерів та методів на основі дерев рішень.

Обробка пропущених даних є критично важливою в таких сферах:

1. Медицина: У медичних дослідженнях та клінічних випробуваннях пропущені дані можуть бути серйозною проблемою для інтерпретації результатів. Імпутація допомагає зберегти важливі медичні дані та підвищити точність діагнозів.

2. Фінанси: У фінансових даних часто зустрічаються пропуски через неповні записи про транзакції або відсутність історичних даних. Правильна імпутація дозволяє покращити точність моделей для прогнозування ризиків та аналізу ринкових тенденцій.

3. Соціальні науки: Пропущені відповіді у соціологічних опитуваннях або переписах населення можна імпутувати для підвищення надійності статистичних висновків.

Таким чином, обробка пропущених даних є ключовим елементом у сучасній аналітиці даних. Завдяки використанню глибинного навчання та складних моделей нейронних мереж, таких як згорткові нейронні мережі, можна досягти нових висот у точності імпутації пропусків, що в свою чергу підвищить ефективність машинного навчання в різних сферах застосування.

1.2 Аналіз відомих методів обробки пропущених даних

У цьому підрозділі розглядаються основні роботи, присвячені методам обробки пропущених даних. Оскільки наявність пропущених даних є поширеною проблемою у багатьох програмах, орієнтованих на дані, досліджено безліч підходів для вирішення цієї проблеми. Зокрема, імпутація (заміщення) пропущених значень зазвичай вимагає певних припущень щодо розподілу даних. Якщо ці припущення неправильні, це може призвести до упередженості в оцінках заміщених значень. Традиційні методи імпутації пропущених даних можна розділити на дві основні категорії: статистичні методи та методи на основі машинного навчання.

У різних галузях були запропоновані різні підходи до імпутації для вирішення проблеми пропущених даних. Наприклад, існують такі методи, як підхід на основі центра класів [41], KNN [13], нечітке кластерування [25], метод "bagging" [2], автоенкодери [15, 27], правила схожості [37], метод множинної імпутації за допомогою ланцюгових рівнянь (MICE) [46], та методи на основі випадкових лісів [39].

Наприклад, у роботі [41] було запропоновано підхід імпутації пропущених значень на основі центру класів. Цей метод модифікував традиційну імпутацію за середнім значенням, використовуючи відстань між кожною точкою даних і центром класу. Порогове значення для імпутації визначалося на основі спостережуваних даних.

У [8] використовувались різні моделі на основі машинного навчання, такі як KNN, метод опорних векторів (SVM) і дерева рішень для заміщення пропущених значень через формулювання проблеми імпутації як задачі оптимізації. KNN вважається одним з найбільш популярних методів завдяки своїй простоті та ефективності.

У роботі [13] було запропоновано розширення традиційного KNN у вигляді методу "чистоти k найближчих сусідів" (PkNNI), де чистота запису розраховується на основі голосів сусідів, обраних як найближчі. Однак, для великих наборів даних цей метод може бути дуже ресурсоємним, оскільки для кожного неповного запису потрібно перевірити всі інші записи. Це робить метод обмеженим у продуктивності при роботі з великими обсягами пропущених даних [35]. Крім того, пошук найближчих сусідів і правильного методу для розрахунку відстані може бути складним завданням [4].

У роботі [39] був запропонований метод імпутації на основі випадкових лісів, який показав, що його ефективність можна підвищити за рахунок збільшення кореляції між ознаками. У методі MIAEC [46] ланцюговий підхід використовувався для того, щоб зібрати всі докази, пов’язані з пропущеними значеннями, і налаштувати подальші оцінки цих значень. Також у роботі Тянь та ін. [40] була застосована техніка нечіткого кластерування FCM для групування спостережуваних даних з метою навчання класифікатора на основі теорії сірих систем для імпутації пропущених даних.

У [43] було запропоновано ітеративний підхід під назвою "ітераційна регресійна імпутація атрибутів," де створюється послідовність регресійних моделей для поступового заповнення пропущених значень.

Традиційні методи імпутації пропущених даних здебільшого фокусуються на використанні різних ймовірнісних моделей або регресійних методів для заповнення пропусків, використовуючи обмежену кількість інформації. Вони часто не можуть забезпечити точну імпутацію, особливо для даних із високим рівнем пропусків. Однак для великих наборів даних методи імпутації на основі машинного навчання (наприклад, KNN, MICE та нечітке кластерування)

потребують багато обчислювальних ресурсів, що ускладнює їх використання для великих обсягів пропущених даних.

Останнім часом, зі зростанням обсягів даних, глибинне навчання показало великий потенціал у різних галузях, таких як біологія [45], реконструкція зображень [6, 50], біомедична візуалізація [12], та геномні дослідження [10]. Було також запропоновано кілька методів глибинного навчання спеціально для імпутації пропущених даних, що дало обнадійливі результати [5, 24].

Наприклад, у роботі [36] використовувалась багатошарова перцептронна (MLP) мережа для імпутації пропущених значень, і було досліджено вплив різних правил навчання та параметрів моделі на кінцеву продуктивність.

У роботах [15, 27] архітектура автоенкодера була використана для відновлення пропущених значень шляхом мінімізації помилки реконструкції.

У [47] для імпутації пропущених даних було використано GAN. Однак ці методи вимагають великих обчислювальних ресурсів для навчання моделі.

У [1] запропонували гібридний підхід на основі нейронних мереж та генетичних алгоритмів для імпутації пропущених даних у медичних системах IoT. Цей підхід використовує переваги глибинного навчання для заповнення пропущених даних та ефективність генетичних алгоритмів для оптимізації ваг нейронної мережі.

У роботі [27] також було запропоновано модель імпутації на основі автоенкодера, яка знижує складність даних. Крім того, у MIDA [23] запропонували модель на основі глибокого автоенкодера для роботи з різними типами даних, різними патернами відсутності значень та пропорціями пропусків у даних.

У роботі [20] було запропоновано підхід для імпутації пропущених значень на основі глибинного навчання для обробки даних про дорожній рух. Цей підхід може виявляти кореляції в даних шляхом пошарового попереднього навчання і поліпшувати результати імпутації завдяки подальшому точному налаштуванню.

У [9] використовувалася глибока модель для імпутації пропущених значень у часових рядах, де модель захоплювала довгострокові тимчасові залежності спостережень і використовувала патерни пропущених значень для

покращення точності передбачень завдяки маскуванню та обліку часових інтервалів у глибокій моделі.

Крім того, у [49] було запропоновано новий метод локальної схожості для імпутації даних, який оцінював значення для заповнення пропусків на основі швидкого кластерування та вибору найближчих сусідів. Зокрема, використовували двошаровий автоенкодер, комбінований із характерною імпутацією для визначення основних ознак набору даних у процесі кластеризації.

Зазначені методи спрямовані на прогнозування значень для заповнення пропущених даних поодинці за допомогою кластеризації, регресії або нейронних мереж.

1.3 Аналіз метрик оцінки ефективності моделей машинного навчання

Аналіз метрик, що використовуються в роботі, є ключовим елементом для оцінки продуктивності запропонованого підходу до імпутації пропущених даних та його впливу на класифікацію. Для оцінки якості імпутації та подальшого впливу на моделі машинного навчання використовуються кілька важливих метрик, кожна з яких має свої особливості й призначення.

Розглянемо їх детальніше.

1. Точність класифікації (Accuracy).

Точність класифікації є основною метрикою, яка використовується для оцінки ефективності моделей машинного навчання після імпутації пропущених даних. Вона визначається як частка правильно передбачених класів до загальної кількості передбачень:

$$Accuracy = \frac{\text{Кількість правильно передбачених класів}}{\text{Загальна кількість передбачених класів}} \quad (1.1)$$

Ця метрика особливо важлива для задач, де ціль полягає у класифікації об’єктів, таких як розпізнавання зображень або виявлення аномалій. В роботі CNNI точність класифікації є ключовим показником того, наскільки ефективно

заповнені пропущені дані можуть допомогти покращити результати класифікації. Оскільки основна мета імпутації полягає не лише в точному відновленні пропущених значень, а й у покращенні якості класифікації, саме точність класифікації найкраще відображає загальний успіх підходу.

2. Середня абсолютна помилка (Mean Absolute Error, MAE).

Середня абсолютна помилка (MAE) використовується для оцінки різниці між фактичними значеннями та імпутованими значеннями. MAE вимірює середню величину відхилення імпутованих значень від справжніх значень у наборах даних:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (1.2)$$

де y_i – справжнє значення;

$\hat{y}_i$ – імпутоване значення.

MAE є корисною метрикою для оцінки точності імпутації, оскільки показує середню помилку без врахування її напряму (недооцінка чи переоцінка). У контексті імпутації пропущених значень MAE може бути важливою для порівняння різних підходів до заповнення пропусків. Однак, в цій роботі основна увага приділяється точності класифікації, оскільки низьке значення MAE не завжди означає, що це покращує результати класифікації.

3. Корінь середньоквадратичної помилки (Root Mean Squared Error, RMSE).

RMSE є ще однією метрикою, яка використовується для оцінки якості імпутації. Вона вимірює середнє квадратичне відхилення між фактичними та імпутованими значеннями:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (1.3)$$

На відміну від MAE, RMSE більш чутлива до великих відхилень між справжніми та імпутованими значеннями. Це означає, що RMSE буде більшим,

якщо імпутація має кілька суттєвих помилок, що робить її хорошою метрикою для виявлення "викидів". Незважаючи на це, як і у випадку з MAE, RMSE не є основною метрикою в роботі, оскільки мета полягає в покращенні класифікації, а не тільки в точності імпутації.

4. Крос-ентропійна втрата (Categorical Cross-Entropy Loss).

Крос-ентропія – це одна з найпоширеніших функцій втрат для задач класифікації, особливо в задачах, де результати подаються у вигляді ймовірностей. Ця метрика використовується для порівняння передбачуваних ймовірностей кожного класу з фактичними мітками класів. Формула крос-ентропії виглядає так:

$$CrossEntropy = -\sum_{i=1}^{n} y_i \log(p_i) \quad (1.4)$$

де y_i – справжня мітка (одиниця для правильного класу, нуль для інших класів);

p_i – ймовірність, що модель приписує цьому класу.

Крос-ентропія є ключовою функцією втрат для тренування моделей глибокого навчання, оскільки допомагає мінімізувати різницю між передбаченими ймовірностями та фактичними результатами. В роботі CNNI ця функція використовується для оптимізації ваг моделі під час імпутації та класифікації, що дозволяє покращити загальну ефективність моделі.

5. Стабільність продуктивності при різних рівнях пропусків/

Окрім загальних метрик, таких як точність класифікації та крос-ентропійні втрати, у роботі також оцінюється стабільність продуктивності запропонованого підходу при різних рівнях пропусків у даних. Цей підхід перевіряється на наборах даних з пропусками від 10% до 70%. Для кожного рівня пропусків аналізується, як змінюється точність класифікації та інші метрики. Висока стабільність моделі навіть при великій кількості пропусків є важливим показником її ефективності.

6. Час обчислення та ресурсні витрати/

Хоча це не є прямою метрикою, у роботі також враховується обчислювальна складність методів імпутації. Наприклад, методи KNN або GAIN можуть бути обчислювально дорогими, особливо для великих наборів даних, оскільки вимагають значного часу для знаходження найближчих сусідів або тренування генеративних мереж. У той час як CNNI, завдяки своїй архітектурі, може працювати швидше на великих масивах даних і бути більш ефективним у контексті великих пропусків.

7. Метрики для порівняння з іншими підходами.

Для повної оцінки запропонованого методу CNNI проводиться порівняння з іншими популярними методами імпутації, такими як GAIN, MICE, KNNI та RandomForest. Використання кількох метрик дозволяє порівняти продуктивність CNNI з цими методами за кількома показниками, забезпечуючи комплексний аналіз. Окрім точності класифікації, аналізується також стабільність кожного методу, обчислювальна ефективність та здатність адаптуватися до різних рівнів пропусків у даних.

Отже, метрики, такі як точність класифікації та крос-ентропія, є ключовими для оцінки успішності імпутації пропущених даних у контексті класифікації. Хоча MAE та RMSE корисні для оцінки точності самого процесу імпутації, метою роботи є покращення класифікації, тому саме класифікаційні метрики є найважливішими. Стабільність продуктивності та порівняння з іншими підходами допомагають зробити висновки про переваги запропонованого методу CNNI, особливо у випадках з великими наборами даних та високим рівнем пропусків.

1.4 Постановка задачі дослідження

У сучасній аналітиці даних і машинному навчанні однією з найбільш поширених і критичних проблем є наявність пропущених значень у наборах даних. Пропуски в даних можуть суттєво вплинути на точність та ефективність моделей, що базуються на цих даних. Втрата частини інформації призводить до підвищеної похибки та зниження продуктивності моделей машинного навчання,

особливо у випадках роботи з великими масивами даних, де просторові і нелінійні взаємозв'язки між ознаками є важливими [51].

Основна мета цього дослідження – розробка ефективного підходу до імпутації пропущених даних, який використовує можливості глибокого навчання, зокрема згорткових нейронних мереж. Пропонований підхід спрямований на покращення якості класифікації шляхом коректного заповнення пропущених значень з використанням просторових і нелінійних властивостей даних.

Основні задачі дослідження:

1. Провести аналіз сучасних підходів до імпутації пропущених даних з метою визначення їх переваг і недоліків у контексті відновлення пропущених значень та точності класифікації.

2. Розробити метод імпутації на основі згорткової нейронної мережі, яка враховує просторові і нелінійні зв’язки у даних для точнішого заповнення пропущених значень.

3. Використати кластеризацію для покращення організації даних та підвищення ефективності навчання моделі.

4. Дослідити, як врахування просторових відносин між ознаками у матриці даних впливає на якість імпутації. Особлива увага приділяється тому, як такі зв’язки можуть бути використані для покращення результатів прогнозування.

5. Дослідити стабільність запропонованого підходу при різних рівнях пропусків (від 10% до 70%) у даних і перевірити, чи впливає зростання кількості пропусків на точність імпутації та подальшої класифікації.

6. Провести серію експериментів оцінки ефективності запропонованого методу у порівнянні з іншими методами імпутації. Оцінити точність класифікації, отриману після імпутації пропущених даних за допомогою різних алгоритмів навчання з учителем.

7. Провести аналіз різних конфігурацій архітектури згорткових нейронних мереж, таких як кількість шарів, розмір фільтра, кількість кластерів, та їхній вплив на точність імпутації і класифікації. Визначити оптимальні налаштування для досягнення найкращих результатів.

Таким чином, головна задача цього дослідження полягає в тому, щоб запропонувати ефективний підхід для імпутації пропущених значень, який здатний працювати з великими наборами даних, забезпечуючи при цьому стабільну продуктивність навіть за умов значних пропусків.

Висновки до розділу 1

1. Обробка пропущених даних є однією з ключових проблем у сучасній аналітиці, оскільки відсутність частини інформації може значно вплинути на точність і продуктивність моделей машинного навчання. Методи імпутації даних, такі як статистичні підходи, машинне навчання та глибинне навчання, мають свої переваги й обмеження залежно від структури і природи даних. Сучасні методи на основі глибокого навчання, зокрема згорткові нейронні мережі та генеративні змагальні мережі, демонструють великий потенціал для точного заповнення пропусків. Важливо враховувати просторові кореляції та складні зв’язки між ознаками, що дозволяє значно покращити якість даних і продуктивність класифікаційних моделей у різних сферах, таких як медицина, фінанси та соціальні науки.

2. Проведено аналіз основних підходів до імпутації пропущених значень, які можна розділити на статистичні методи та методи машинного навчання. Традиційні методи, такі як KNN і MICE, мають обмеження в продуктивності та потребують значних обчислювальних ресурсів, особливо для великих наборів даних. Методи, що базуються на глибинному навчанні, зокрема автоенкодери та GAN, демонструють обнадійливі результати, але також потребують складної обчислювальної інфраструктури. У роботах підкреслюється важливість врахування просторових та нелінійних залежностей між ознаками для точнішої імпутації, що є перспективним напрямом для розробки нових методів.

3. У сучасній аналітиці даних і машинному навчанні однією з найбільш поширених і критичних проблем є наявність пропущених значень у наборах даних. Пропуски в даних можуть суттєво вплинути на точність та ефективність моделей, що базуються на цих даних. Втрата частини інформації призводить до

підвищеної похибки та зниження продуктивності моделей машинного навчання, особливо у випадках роботи з великими масивами даних, де просторові і нелінійні взаємозв'язки між ознаками є важливими. Головна задача цього дослідження полягає в тому, щоб запропонувати ефективний підхід для імпутації пропущених значень, який здатний працювати з великими наборами даних, забезпечуючи при цьому стабільну продуктивність навіть за умов значних пропусків.

2 МЕТОД ВІДНОВЛЕННЯ ВІДСУТНІХ ДАНИХ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

2.1 Підхід до відновлення відсутніх даних на основі алгоритму нечіткої кластеризації та згорткової нейронної мережі

У цьому розділі розглянуто метод імпутації пропущених значень, який розроблено на основі згорткової нейронної мережі. Основна ідея цього підходу полягає у використанні натренованого ядра CNN для заповнення пропущених значень. Ваги ядра визначаються шляхом навчання на основі нелінійностей і просторових властивостей, що існують у даних [52].

У даному підході спочатку знаходяться кореляції між ознаками, щоб найбільш корельовані ознаки розташовувалися близько одна до одної. Таким чином, ядро можна навчити ефективніше, використовуючи інформацію про кореляції, що присутні у даних. В іншому випадку ядро може бути недостатньо добре побудованим.

Другим кроком є застосування алгоритму FCM, щоб організувати записи в різні кластери.

FCM – це Fuzzy C-Means (нечітка кластеризація С-середніх), один із найпоширеніших алгоритмів нечіткої кластеризації. Він використовується для кластеризації даних на основі нечіткої логіки, що дозволяє кожній точці належати до кількох кластерів з певними ступенями членства.

Основні характеристики FCM:

– нечіткі кластери: на відміну від класичної кластеризації, де кожна точка належить до одного кластеру, FCM дозволяє точці належати до кількох кластерів з певними ступенями членства;

– критерій мінімізації: алгоритм намагається мінімізувати функцію, яка залежить від відстані між точками даних і центроїдами кластерів;

– оновлення центроїдів і членства: під час роботи алгоритму FCM постійно оновлює центри кластерів (центроїди) та ступінь членства точок до цих кластерів.

Цей алгоритм застосовується в різних галузях, включаючи обробку зображень, машинне навчання, аналіз даних та інші. Він особливо корисний тоді, коли дані мають нечіткі межі між кластерами.

На наступному кроці записи сортуються у висхідному порядку на основі членства в кластерах. Далі, такий підхід застосовується для заповнення пропущених значень у добре організованих даних. В даній роботі увагу зосереджено на імпутації пропущених значень у даних, які мають нелінійні зв'язки та просторові відносини.

Розглянемо постановку задачі.

Нехай є набір даних, представлений у вигляді матриці $m \times n$, де m – це кількість записів (екземплярів), а n – це кількість ознак (фіч). Оригінальний набір даних X розділяється на дві частини: навчальні дані ($X_{obs}^{t} \in \mathbb{R}^{m \times n}$) і тестові дані ($X_{obs}^{t\prime} \in \mathbb{R}^{m\prime \times n}$), як показано на рисунку 2.1.

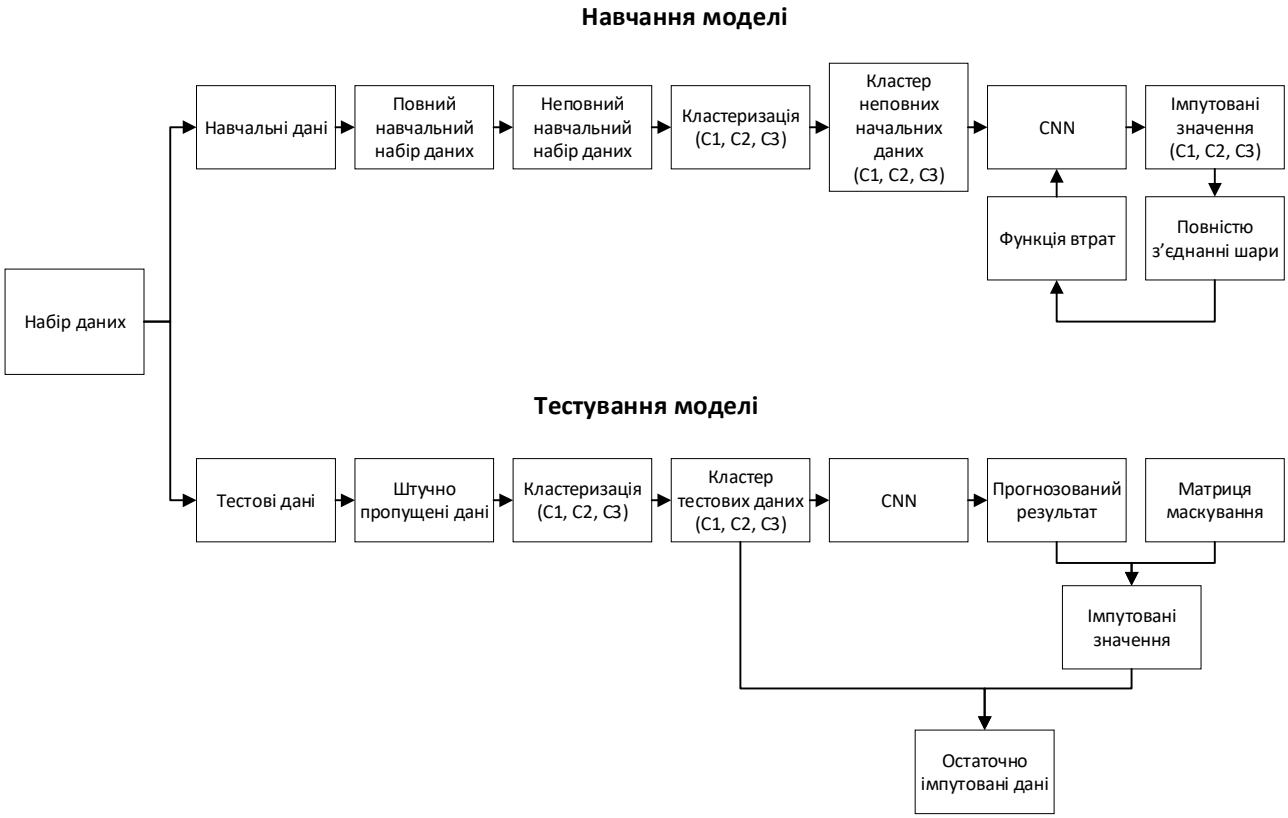


Рисунок 2.1 – Схема підходу імпутації даних

Навчальні дані використовуються для побудови моделі, а тестові – для перевірки її ефективності. Основна мета полягає в тому, щоб ефективно заповнити пропущені значення в даних за допомогою запропонованого підходу,

що базується на CNN, забезпечивши точні результати та покращену якість імпутації.

Після поділу даних для отримання навчального та тестового наборів було штучно створено пропущені значення у $X_{obs}^{t} \in \mathbb{R}^{m \times n}$ та $X_{obs}^{t\prime} \in \mathbb{R}^{m\prime \times n}$, причому ці пропущені значення спочатку заповнені нулями. Після створення пропусків отримано нові версії навчальних і тестових даних, тобто $X_{miss}^{t} \in \mathbb{R}^{m \times n}$ та $X_{miss}^{t\prime} \in \mathbb{R}^{m\prime \times n}$ відповідно. Далі, алгоритм нечіткого кластерування (FCM) застосовується до $X_{miss}^{t} \in \mathbb{R}^{m \times n}$ та $X_{miss}^{t\prime} \in \mathbb{R}^{m\prime \times n}$, щоб розділити дані на різні кластери (як показано на рисунку 2.1).

Основна мета – заповнити пропущені значення, використовуючи просторові відносини та інформацію про нелінійність, яка існує в матриці даних. Тому записи впорядковуються за зростанням на основі значень членства, до якого кластера вони належать, таким чином, щоб записи з ближчими значеннями членства знаходилися поруч, і навпаки. Мета сортування даних полягає в тому, щоб ефективніше навчити модель виявляти нелінійні закономірності з наданих даних.

Алгоритм FCM ефективно організовує n записів $X = \{x_1, \dots, x_n\}$ у кілька нечітких кластерів. Для заданого набору даних FCM повертає k центроїдів кластерів $C = \{c_1, \dots, c_k\}$ і матрицю розбиття W, де кожен елемент $\mu_{i,j}$ представляє ступінь, до якого запис x_i належить j-му кластеру. У даному випадку кількість кластерів $k = 3$. Метою FCM є мінімізація функції мети, як показано в рівнянні:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} (\mu_{i,j})^m \left\| x_i - c_j \right\|^2, \tag{2.1}$$

де $\left\| x_i - c_j \right\|$ – це евклідова відстань між записом x_i і j-м центроїдом кластера.

$m \in \mathbb{R}$ – коефіцієнт нечіткості, який визначає, наскільки нечітким буде кластер. Однак, збільшене значення m призводить до зменшення значення

членства. У цій роботі було встановлено $m = 2$. Ступінь членства запису x_i до j-го кластера обчислюється за рівнянням:

$$\mu_{i,j} = \frac{1}{\sum_{j'=1}^{k} \left(\frac{\|x_i - c_j\|}{\|x_i - c_{j'}\|} \right)^{\frac{2}{m-1}}}, \tag{2.2}$$

де c_j – це центроїд j-го кластера;

k – кількість кластерів.

Визначення центроїду кластера є ітераційним процесом і триває до досягнення умов завершення:

$$c_j = \frac{\sum_{i=1}^{n} (\mu_{i,j})^m x_i}{\sum_{i=1}^{n} (\mu_{i,j})^m}, \forall j = 1,2, \dots, k, \tag{2.3}$$

У загальному випадку сингулярність у знаменнику рівняння (2.2) не виникає, якщо тільки запис x_i не дорівнює центроїду кластера. За рівнянням (2.3), для j-го кластера запис x_i може бути рівний центроїду c_j, коли $\mu_{i,j} = 1$ і $\forall\, i' \neq i : \mu_{i',j}$ для всіх інших кластерів. Однак, це рідко зустрічається в реальних додатках, тому ймовірність отримання нуля у знаменнику рівняння (2.2), що призводить до сингулярності, дуже мала.

Як показано на рисунку 2.1, один блок представляє навчання моделі, а інший – тестування моделі. У блоці навчання маємо чисті навчальні дані $X_{obs}^{t} \in \mathbb{R}^{m \times n}$, до яких штучно створили пропущені значення, і ці пропуски спочатку заповнюються нулями. Після цього отримуємо "забруднений" навчальний набір даних, позначений як $X_{miss}^{t} \in \mathbb{R}^{m \times n}$, на який застосовується алгоритм нечіткого кластерування FCM.

Як видно з рисунку 2.1, центроїди c_1, c_2 та c_3 представляють три кластери, що містять записи з пропущеними даними, і ці кластери передаються до моделі CNN для імпутації пропущених значень згідно з алгоритмом.

Алгоритм 1: навчання запропонованого методу.

Вхід: набір даних X, що має m записів та n ознак.

Вихід: навчена модель CNN.

Крок 1:

1.1. Розділити X на навчальні дані $X_{obs}^{t} \in \mathbb{R}^{m \times n}$ та тестові дані $X_{obs}^{t'} \in \mathbb{R}^{m' \times n}$.

1.2. У $X_{obs}^{t} \in \mathbb{R}^{m \times n}$ випадково видалити деякі значення, щоб створити пропущені дані $X_{miss}^{t} \in \mathbb{R}^{m \times n}$.

1.3. Алгоритм FCM використовується для поділу $X_{miss}^{t} \in \mathbb{R}^{m \times n}$ на k кластерів, а значення членства $\mu_{i,j}$ для кожного запису x_i, що належить j-му кластеру, обчислюється за допомогою рівняння (2.2).

1.4. Знайти кореляції між ознаками $X_{miss}^{t} \in \mathbb{R}^{m \times n}$ і розташувати записи у порядку зростання на основі значень членства, до якого кластера належать ці записи.

Крок 2:

2.1. Пропущені значення $X_{miss}^{t} \in \mathbb{R}^{m \times n}$ передаються до моделі CNN.

2.2. Модель CNN імпутує пропущені значення $X_{miss}^{t} \in \mathbb{R}^{m \times n}$.

2.3. Імпутовані дані передаються до повністю з'єднаних (FC) шарів для передбачення міток записів, використовуючи функцію softmax.

2.4. Обчислити втрати за допомогою визначеної функції втрат (на основі перехресної ентропії).

2.5. Повторювати процедуру, доки втрати не мінімізуються.

Крок 3:

3.1. Ваги ядра оптимізуються, і модель навчається мінімізувати втрати.

3.2. Повернути навчену CNN.

З іншого боку, на етапі тестування, щоб створити пропуски в чистих тестових даних $X_{obs}^{t'} \in \mathbb{R}^{m' \times n}$, повторюється та сама процедура навчання, і тоді отримуємо "забруднені" тестові дані $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$. Після навчання моделі CNN (див. алгоритм 1), вона оцінюється на $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$ для імпутації пропущених

значень, як показано на рисунку 2.1 та в алгоритмі 2. Пропущені значення у $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$ можуть бути імпутовані способом, проілюстрованим у настпних рівняннях:

$$x_{ij} = f_n\left(x_{miss}^{t'}\right), \tag{2.4}$$

$$f_n = f(\sum w_i x_i + b), \tag{2.5}$$

де $f(.)$ – це функція CNN, що виконує згортку над даними;

x_i – це i-й запис;

w_i – вектор ваг;

b – це зсув (bias);

а f – це функція активації.

Алгоритм 2: імпутація та процедура тестування.

Вхід: модель CNN, тестові дані з пропущеними значеннями $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$, що мають m' записів і n ознак.

Вихід: імпутовані дані, що містять m' записів і n ознак.

Крок 1:

1.1. for $i = 1$ до n do

1.2. З тестових даних $X_{obs}^{t'} \in \mathbb{R}^{m' \times n}$ випадково видалити деякі значення, щоб створити тестові дані з пропущеними значеннями $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$.

1.3. Використовуйте алгоритм FCM для організації $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$ у k кластерів. Значення членства μ_{ij}, що визначає, до якого кластеру x_i належить, обчислюється за допомогою рівняння (2.2).

1.4. Впорядкуйте записи у $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$ у порядку зростання на основі значень членства, до якого кластеру вони належать.

Крок 2:

2.1. Імпутуйте $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$ за допомогою рівняння (2.4).

2.2. Повторюйте процедуру, доки всі пропущені значення не будуть імпутовані.

Крок 3:

3.1. Накладіть маску, щоб отримати імпутовані значення ($X_{imputed}^{t'} \in \mathbb{R}^{m' \times n}$) і об'єднайте їх з $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$ щоб отримати фінальні імпутовані дані.

3.2. Використовуйте функцію втрат (на основі перехресної ентропії) для обчислення втрат.

3.3. Виведіть результат втрат.

Можна помітити, що під час виконання операції згортки всі значення даних змінюються. Так само і в нашому випадку, після завершення процесу імпутації, отримуємо нову матрицю даних, у якій всі значення змінені через операцію згортки. Тому, щоб витягти імпутовані значення, до новоутвореної матриці даних застосовується маскування, як показано на рисунку 2.1.

Матриця маскування – це матриця, яка використовується для позначення або виділення певних елементів даних, які потрібно обробити, ігнорувати або замінити в певних обчисленнях. Її елементи мають значення 0 або 1.

Пізніше ці замасковані значення об'єднуються з $X_{miss}^{t'} \in \mathbb{R}^{m' \times n}$, щоб отримати остаточно імпутовані дані. Наприкінці, після маскування, отримуємо повну матрицю з остаточно імпутованими значеннями, як показано на рисунку 2.1.

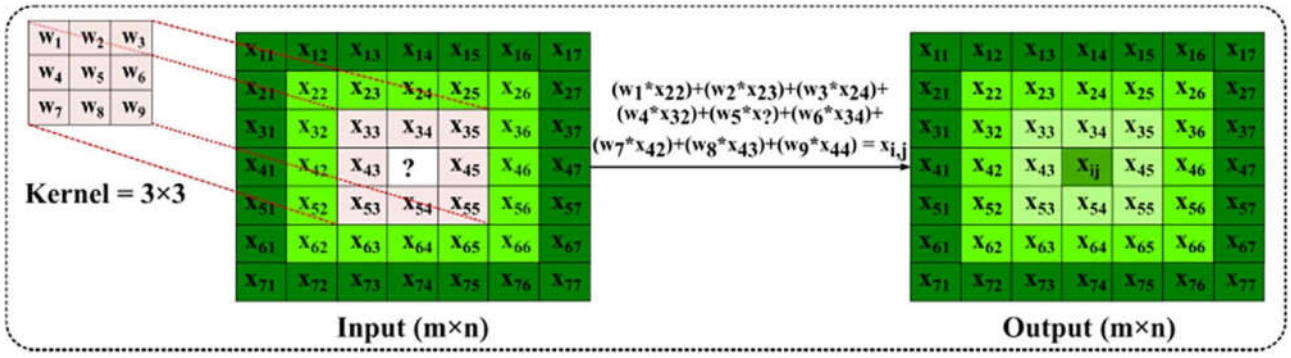
Крім того, рисунок 2.2 ілюструє приклад того, як можна імпутувати одне значення за допомогою ядра для вхідних даних $m \times n$.

На етапі попередньої обробки спочатку визначаємо кореляції, що існують у матриці даних, щоб найкорельованіші значення ознак були розташовані поруч. Рисунок 2.2(a) представляє початкове розташування даних перед виявленням просторових зв'язків між значеннями ознак, тоді як на рисунку 2.2(b) різними кольорами показано різні ступені просторового зв'язку відомих значень із пропущеним значенням після просторового впорядкування даних через виявлення просторових зв'язків між значеннями ознак. У цьому новому

впорядкуванні даних ядро може бути навчене ефективніше, використовуючи інформацію про кореляції, що існують у наданих даних.

x_{11}	x_{15}	x_{12}	x_{14}	x_{17}	x_{13}	x_{16}
x_{21}	x_{25}	x_{22}	x_{24}	x_{27}	x_{23}	x_{26}
x_{41}	x_{45}	x_{42}	x_{44}	x_{47}	x_{43}	x_{46}
x_{71}	x_{75}	x_{72}	?	x_{77}	x_{73}	x_{76}
x_{51}	x_{55}	x_{52}	x_{54}	x_{57}	x_{53}	x_{56}
x_{31}	x_{35}	x_{32}	x_{34}	x_{37}	x_{33}	x_{36}
x_{61}	x_{65}	x_{62}	x_{64}	x_{67}	x_{63}	x_{66}

а)



б)

Рисунок 2.2 – Приклад процесу імпутації пропущених значень:
а) початкове розташування даних до визначення просторових взаємозв'язків між значеннями ознак;
б) просторове розташування даних після виявлення просторових взаємозв'язків між значеннями ознак, де різні кольори використовуються для позначення різного ступеня просторового зв'язку відомих значень з пропущеним значенням

Нехай є ядро розміром $M \times N$ (де M – це кількість рядків, а N – кількість стовпців), і значення одиниці x_{ij} у карті ознак відсутнє. Значення x_{ij} обчислюється як зважена сума вхідних даних, які містяться у фрагменті розміром $M \times N$, як показано на рисунку 2(b). Ваги ядра є навчуваними параметрами, які оцінюються шляхом використання інформації про нелінійності з наданих даних. Після того, як ваги оптимізовано на етапі навчання, їх можна використовувати

для імпутації пропущених даних на етапі тестування. Рівняння (2.6) показує, як фільтр виконує згортку по даних та імпутує пропущене значення x_{ij}.

$$x_{ij} = (W * X)(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} X_{i-m,j-n} W_m n, \quad (2.6)$$

де символ $*$ – позначає операцію згортки;

m та n – використовуються для позначення індексів рядків і стовпців, відповідно, тобто $m = 1$ до M, а $n = 1$ до N.

Крім того, *i* та *j* представляють індекси рядка і стовпця, в яких знаходиться пропущене значення. Однак після завершення тестування ми отримуємо нову матрицю даних, у якій всі значення змінені через операцію CNN. Імпутовані дані витягуються шляхом застосування маски на частини з пропусками у фінальній імпутованій матриці, як показано на рисунку 2.1. Пізніше ці замасковані значення об'єднуються з пропущеними через те, що пропущені значення замінюються новоімпутованими даними, а решта даних залишається незмінною (див. рисунок 2.1).

2.2 Архітектура згорткової нейронної мережі

Запропонований метод використовує певний тип архітектури нейронних мереж, відомий як згорткова нейронна мережа. На рисунку 2.3 показано архітектуру мережі, і можна побачити, що розроблена архітектура складається з щільно пов’язаної згорткової нейронної мережі з резидуальними (залишковими) з’єднаннями (CNN-DR).

Резидуальні з’єднання (англ. residual connections), або залишкові зв’язки – це механізм у нейронних мережах, який дозволяє передавати інформацію (вихід попереднього шару) до наступного шару без змін або з мінімальними модифікаціями. Вперше цю ідею запропонували у ResNet (Residual Network), одній з найбільш популярних архітектур глибинного навчання.

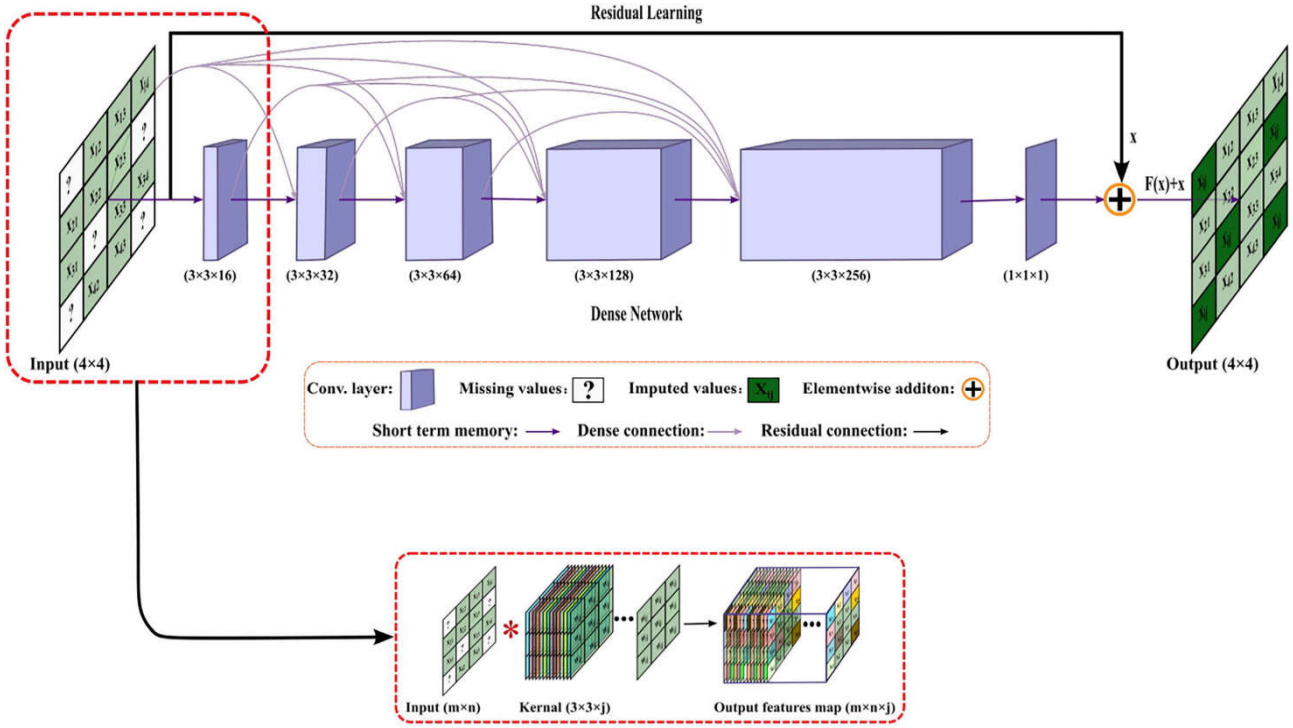


Рисунок 2.3 – Архітектура згорткової нейронної мережі для імпутації пропущених даних

Оскільки розроблена архітектура має просту схему з'єднань, вона спрямована на забезпечення максимально можливого потоку інформації між шарами в мережі. Тому всі шари щільно з'єднані безпосередньо один з одним. Проте, щоб зберегти спрямовану передачу даних, кожен шар отримує додаткові вхідні дані від усіх попередніх шарів і передає свої карти ознак усім наступним шарам. У щільному з'єднанні використовується конкатенація для отримання колективних знань від усіх попередніх шарів.

Щільна мережа є типом згорткової нейронної мережі, яка використовує щільне з'єднання між шарами, де кожен шар безпосередньо підключений до всіх інших шарів у напрямку вперед. На відміну від традиційної згорткової мережі, де кожен шар підключений лише до свого наступного шару (наприклад, L шари мають L наборів з'єднань), щільне з'єднання вирішує проблему зникнення градієнта, посилює передачу ознак і сприяє повторному використанню ознак.

Щільні та резидуальні з'єднання виконують різні операції: у щільному з'єднанні ми об'єднуємо всі попередні карти ознак через конкатенацію. З іншого боку, у резидуальному з'єднанні карти ознак об'єднуються за допомогою

додавання перед тим, як вони передаються до певного шару. Перевага наявності цих двох типів з'єднань (щільного і резидуального) полягає в тому, що в резидуальному з'єднанні можна використовувати лише одну попередню карту ознак, тоді як у щільному з'єднанні можна використовувати ознаки, отримані від усіх попередніх згорткових блоків.

Як видно на рисунку 2.3, запропонована архітектура CNN-DR має шість 2D згорткових шарів з поступово збільшуваною кількістю фільтрів і одним резидуальним з'єднанням. На рисунку 3(a) суцільна лінія, яка передає вхідний шар до оператора додавання, відома як резидуальне з'єднання. Зокрема, у найкращому випадку додаткові шари глибокої нейронної мережі можуть працювати краще, ніж використання менш глибокої мережі, і також значно зменшити похибку. Тому було додано резидуальний блок для досягнення кращої продуктивності порівняно з простою глибокою нейронною мережею. Цей блок представлений, як показано в рівнянні

$$x_{i+1} = x_i + F(x_i, W_i), \tag{2.7}$$

де x_{i+1} і x_i – вихідні та вхідні значення мережі відповідно;

F – це резидуальна функція;

W_i – представляє параметри блоку.

Наступне рівняння показує функцію перетворення ознак

$$H(x) = F(x) + x, \tag{2.8}$$

де x – це вхід.

Аналогічно, наступне рівняння показує резидуальну функцію:

$$F(x) = relu(w_2 * (relu(w_1 x + b))), \tag{2.9}$$

де $relu$ – це функція активації;

w_1 і w_2 –ваги в першому та другому шарах;

b – зсув.

Було використано резидуальний блок у простій мережі, оскільки встановлення пропускного з'єднання в резидуальній мережі вирішує проблему зникнення градієнта. Зокрема, ця проблема вирішується шляхом надання альтернативного шляху через резидуальний блок, через який може протікати градієнт. Також це з'єднання допомагає моделі гарантувати, що вищі шари будуть працювати щонайменше так само добре, як і нижчі шари. З резидуальним блоком вхідні дані можуть швидше передаватися через з'єднання між шарами.

Як видно на рисунку 3(а), вхідна матриця даних має випадково створені пропущені значення, і на виході є відповідна повна матриця даних. Щоб уникнути втрати інформації та зберегти консистентність розміру даних, використовується додаткове заповнення (padding), щоб розмір вихідних даних залишався таким же, як і вхідний. Крім того, у розробленій архітектурі ядра мають деякі ваги, які є параметрами навчання, і ці ваги формуються на основі навколишніх значень, що потрапляють під ядро. Також у якості функції активації $relu$ через її ефективність і популярність у практичних додатках. Параметри навчання в згортковому шарі оцінюються за наступним рівнянням:

$$(K_h * K_h * F_{i-1} + 1) * F_i, \tag{2.10}$$

де $K_h \times K_h$ – це розмір згорткового ядра;

F_{i-1} – кількість карт ознак у попередньому та поточному шарах;

+1 – вказує на те, що додається зсув.

Крім того, на рисунку 2.3 показано імпутацію випадково пропущеного значення вибірки даних, яка проходить через модель навчання. Після того, як модель навчена, вона здатна імпутувати пропущені значення за допомогою натренованого ядра.

Запропонований підхід розглядається як схема спільного навчання "кінець-в-кінець", де імпутація пропущених даних і класифікація виконуються одночасно. Іншими словами, запропонований підхід не зосереджується виключно на імпутації пропущених значень, оскільки основною метою є

покращення продуктивності класифікації, а імпутовані дані повинні сприяти підвищенню ефективності навчання класифікатора, як зазначено в роботах [21, 25].

Крім того, оскільки справжнє значення, яким слід замінити пропущене, часто невідоме в реальному житті [30], функція втрат була розроблена без потреби знати справжнє значення. Вона використовується для оцінки того, наскільки ефективно імпутуються пропущені значення з точки зору покращення продуктивності в задачах класифікації.

Що стосується розробки запропонованої функції втрат, ми використовуємо повністю з'єднані шари для класифікації імпутованих даних, де функція активації softmax використовується для виведення вектору ймовірностей класів для кожного з імпутованих записів.

У цій конфігурації справжні мітки $X_{true}^{L} \in \mathbb{R}^{m \times n}$ і прогнозовані $X_{pred}^{L} \in \mathbb{R}^{m \times n}$ передаються до функції втрат для обчислення втрат, де справжні мітки перетворюються у one-hot вектори, які можна безпосередньо порівнювати з прогнозованими ймовірнісними розподілами. Зокрема, різниця між $X_{true}^{L} \in \mathbb{R}^{m \times n}$ і $X_{pred}^{L} \in \mathbb{R}^{m \times n}$ обчислюється за допомогою категоріальної перехресної ентропії, як показано в рівнянні:

$$CrossEntropy = -\sum_{i=1}^{n} X_{true(i)}^{L} \log\left(X_{pred(i)}^{L}\right), \tag{2.11}$$

де n – загальна кількість класів;

$X_{true(i)}^{L}$ – індикатор того, чи є i-й клас справжнім;

$X_{pred(i)}^{L}$ – ймовірність, оцінена для i-го класу.

Щоб зменшити втрати за перехресною ентропією, ваги ядра CNN змінюються таким чином, щоб зменшити різницю між прогнозованими та справжніми мітками. Ця процедура триває, доки втрати не будуть мінімізовані (як показано на рисунку 2.1).

Висновки до розділу 2

1. Розглянуто метод імпутації пропущених значень на основі згорткових нейронних мереж і алгоритму нечіткої кластеризації FCM. Даний підхід використовує натреноване ядро CNN, яке виявляє просторові та нелінійні залежності між ознаками, для заповнення пропусків у даних. Алгоритм FCM допомагає ефективно організувати дані в кластери, покращуючи точність імпутації. Таким чином, цей підхід дозволяє покращити якість заповнення пропущених даних, що в свою чергу підвищує ефективність подальшого аналізу та класифікації.

2. Запропоновано архітектуру згорткової нейронної мережі, яка використовує щільні та резидуальні з'єднання для ефективної імпутації пропущених даних. Щільне з'єднання дозволяє кожному шару передавати інформацію всім наступним, забезпечуючи максимальне використання ознак. Резидуальні з'єднання допомагають вирішити проблему затухання градієнта, зберігаючи продуктивність глибокої нейронної мережі на високому рівні. Архітектура з шістьма згортковими шарами дозволяє ефективно заповнювати пропущені значення, що покращує якість даних та кінцеві результати моделі.

3. Розглянуто використання функції втрат у рамках даного підходу, який поєднує імпутацію пропущених даних та класифікацію. Основна мета функції втрат полягає в підвищенні продуктивності класифікації, оскільки справжнє значення пропущених даних зазвичай невідоме. Використання категоріальної перехресної ентропії дозволяє порівнювати справжні та прогнозовані мітки, а зміна ваг згорткової мережі спрямована на мінімізацію цих втрат. Таким чином, спільне навчання моделі підвищує точність як імпутації пропущених значень, так і кінцевої класифікації.

3 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА РЕЗУЛЬТАТИ ОЦІНКИ ЕФЕКТИВНОСТІ ІМПУТАЦІЇ ДАНИХ

3.1 Постановка експериментальних досліджень

У цьому розділі представлено експериментальні дослідження запропонованого методу та оцінено, наскільки суттєва перевага імпутації пропущених значень за допомогою даного підходу порівняно з іншими методами. Зокрема, було обрано 7 наборів даних з репозиторію машинного навчання UCI, деталі яких наведено нижче. Спочатку на чистих записах було штучно створено випадкові пропуски, і ці пропущені значення були замінені на нулі. Пізніше ці пропущені значення були імпутовані за допомогою запропонованого підходу, а також було відібрано кілька інших технік імпутації для порівняння, а саме KNNI [7], MissForest [38], MICE [33], GAIN [47]. На кінець, було підтверджено стабільність запропонованого підходу за різних пропорцій пропущених даних (тобто 10%, 20%, 30%, 40%, 50%, 60%, 70%).

Оцінка ефективності здійснювалася за допомогою п'яти алгоритмів навчання з учителем (логістична регресія (LR), лінійний дискримінантний аналіз (LDA), KNN, наївний Байєс (NB), багатошаровий перцептрон (MLP)) для навчання класифікаторів на імпутованих даних. Для кожного набору даних експеримент проводився 5 разів у випадкових умовах поділу даних (80% для навчання та 20% для тестування), і середня точність класифікації разом зі стандартним відхиленням використовувалася як кінцевий результат імпутаційної продуктивності. Спочатку було штучно видалено 20% значень атрибутів у всіх записах з чистих даних. У результаті було отримано набір даних із пропущеними значеннями.

3.2 Набори даних та налаштування параметрів

У таблиці 3.1 наведено деталі обраних наборів даних, такі як кількість атрибутів, кількість записів та кількість класів.

Таблиця 3.1 – Обрані набори даних UCI для експериментальної роботи

Набір даних	Кількість записів	Кількість атрибутів	Кількість класів	Пропущені дані
Letter Recognition (LR)	20,000	16	26	Ні
Optdigits (OD)	5,620	64	10	Ні
Segmentation (Sg)	2,310	19	7	Ні
Pendigits (PD)	10,992	16	10	Ні
Parkinson (Pk)	5,875	26	42	Ні
Ozone (Oz)	2,536	73	2	Так
Waveform (WF)	5,000	21	3	Ні

З таблиці 3.1 можна побачити, що жоден з наборів даних, крім набору Ozone, не містить пропущених значень. Однак, для оцінки ефективності імпутації пропущених значень потрібно, щоб набори даних не містили пропущених значень, щоб можна було провести пряме порівняння з результатами, отриманими на чистих даних. Тому на етапі попередньої обробки всі записи, які мали початкові пропуски, були видалені. Натомість було штучно видалено деякі значення ознак із цих чистих записів для створення пропущених значень.

Крім того, кожне значення було нормалізоване до діапазону [0, 1] для того, щоб уникнути руйнівного впливу великих значень при обчисленнях, тобто нормалізація була застосована для збереження більшої узгодженості між доменами ознак.

Запропонована архітектура на основі CNN показана на рисунку 2.3. Мережа складається з шести 2D-згорткових шарів, кожен з яких має послідовно зростаючу кількість ядер: 16, 32, 64, 128 і 256 відповідно, з розміром 3×3, як

показано на рисунку 2.3. Швидкість навчання [26] встановлена на рівні $1e^{-3}$, а розмір пакету (batch size) – 4. Оптимізатор Adam використовується з параметрами за замовчуванням для мінімізації функції втрат (з огляду на високу ефективність цього оптимізатора). Для обчислення втрат використовується індивідуальна функція втрат – перехресна ентропія.

Крім того, для навчання мережі потрібно визначити кількість епох. Оскільки занадто мала кількість епох може призвести до недонавчання, а велика кількість епох – до перенавчання та високих обчислювальних витрат, у даному експерименті визначено 500 епох. Цікаво, що операція згортки у CNN зменшує розмірність вхідних даних, і існують певні граничні умови для елементів, що знаходяться поблизу граничних ліній. Щоб вирішити цю проблему, було використано заповнення шляхом додавання нулів до кінця вхідного масиву, а розмір кроку встановлено в 1. Перевагою додавання такого заповнення є те, що розмір вихідних даних залишається таким самим, як і вхідний. Крім того, для кожного шару використовується функція активації ReLU.

Приклад реалізації згорткової нейронної мережі для задачі імпутації пропущених даних, з урахуванням вище зазначених параметрів представлено в додатку А.

3.3 Критерії оцінки ефективності

У цій роботі в якості метрики для оцінки запропонованого підходу було обрано точність класифікації. Продуктивність імпутації оцінюється за допомогою п’яти відомих алгоритмів навчання з учителем, а саме: логістична регресія (LR), лінійний дискримінантний аналіз (LDA), KNN, наївний Байєс (NB) та багатошаровий перцептрон (MLP). Точність класифікації вимірюється для визначення ефективності імпутації пропущених даних щодо продуктивності класифікації. Важливо зазначити, що метою експериментів є аналіз того, наскільки ефективно імпутація пропущених значень допомагає покращити продуктивність класифікації. Тому вибір таких метрик, як середня абсолютна помилка (MAE) та корінь середньоквадратичної помилки (RMSE), не є

доцільним для оцінки продуктивності, оскільки нижчі значення MAE або RMSE для імпутації не обов’язково вказують на вищий показник точності класифікації.

З цієї причини було обрано різні алгоритми для навчання класифікаторів для оцінки продуктивності у відсутності пропущених значень і в їхній присутності. Було розглянуто два аспекти для оцінки продуктивності запропонованого підходу. По-перше, потрібно проаналізувати, наскільки добре навчання на імпутованих даних порівнюється з навчанням на чистих даних. По-друге, потрібно підтвердити ефективність даного методу, порівнюючи його з іншими. Для оцінки, з цих точок зору, оригінальний набір даних поділяється на 3 частини: чистий навчальний піднабір, піднабір з пропущеними значеннями та тестовий набір (пропорція поділу становить 60%, 20%, 20% відповідно).

Процедуру оцінки продуктивності було проводено у два етапи. На першому етапі використано зазначені вище алгоритми для навчання класифікаторів на повністю чистих даних і проведено оцінку кожного класифікатора на тестових даних (без пропущених значень). На другому етапі використано ті самі алгоритми для навчання класифікаторів на суміші чистих записів і записів з імпутованими значеннями, щоб оцінити їх за допомогою тестових даних (із деякими пропущеними значеннями).

У таблицях в додатку Б наведено результати, отримані на цих двох етапах.

3.4 Результати експериментальних досліджень

У цьому розділі представлено детальний опис порівняння запропонованого методу з іншими методами імпутації пропущених значень. У таблицях додатку Б показана точність класифікації, отримана за допомогою різних методів на випадково згенерованих пропущених значеннях. Як видно з даних таблиць, існує значна різниця у продуктивності між запропонованим методом та іншими підходами. Цю різницю можна візуалізувати на рисунках 3.1-3.5, які є графічним відображенням таблиць додатку Б.

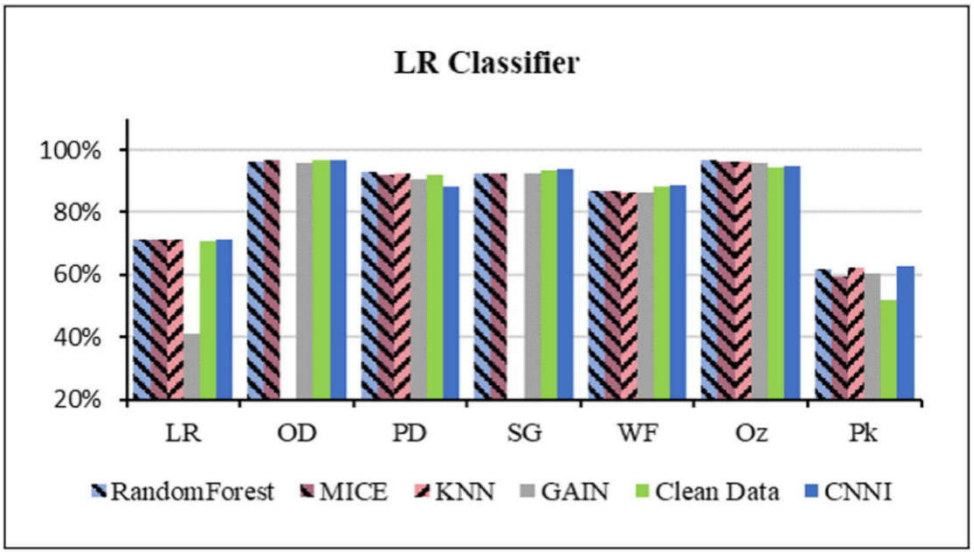


Рисунок 3.1 – Аналіз продуктивності методів імпутації для класифікатора LR

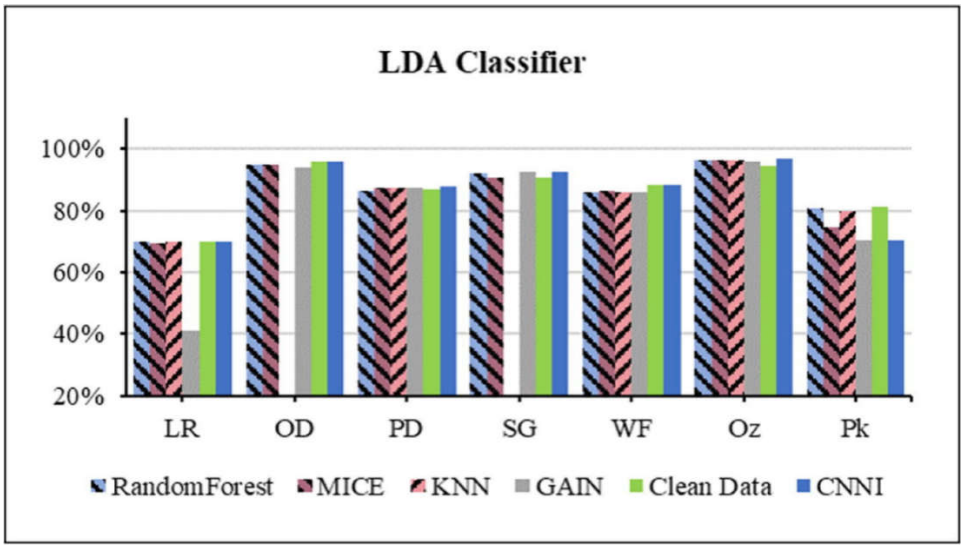


Рисунок 3.2 – Аналіз продуктивності методів імпутації для класифікатора LDA

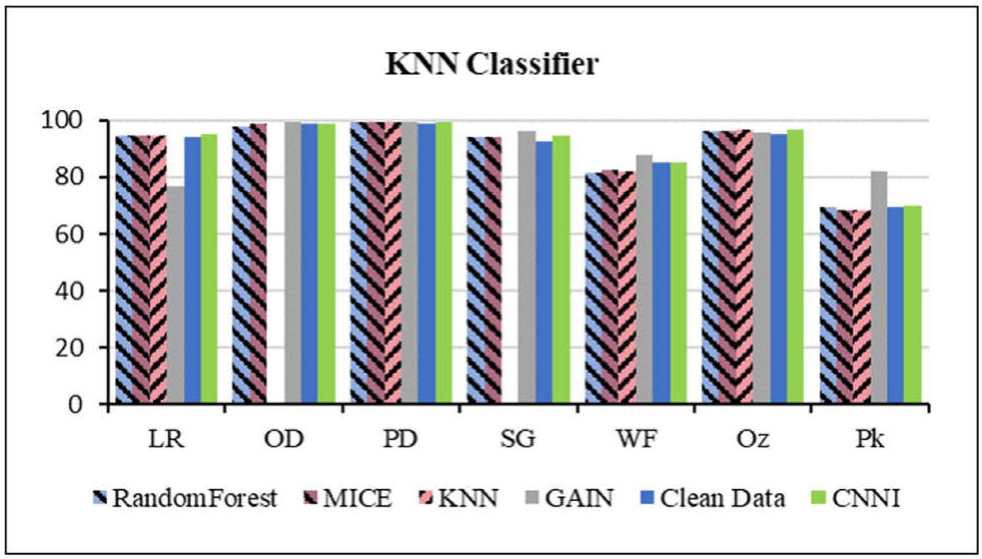


Рисунок 3.3 – Аналіз продуктивності методів імпутації для класифікатора KNN

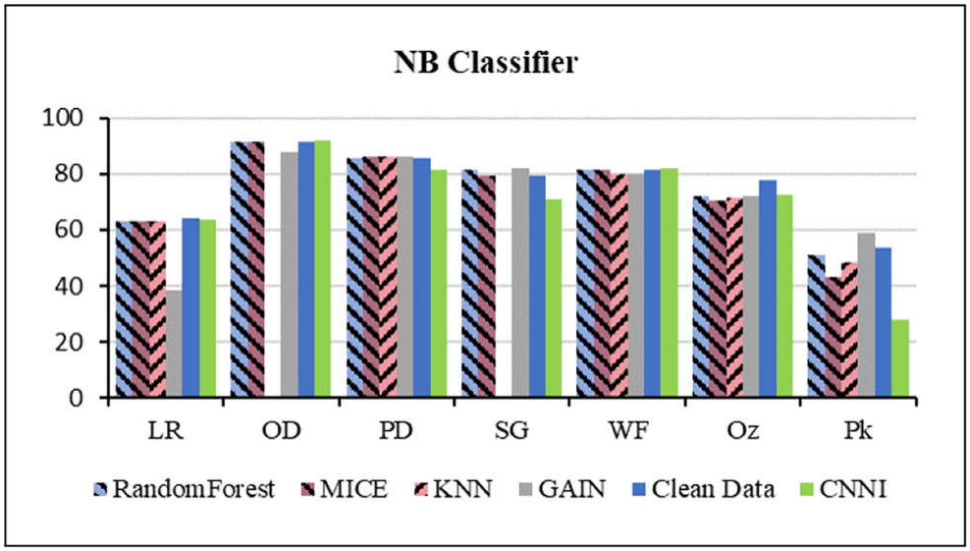


Рисунок 3.4 – Аналіз продуктивності методів імпутації для класифікатора NB

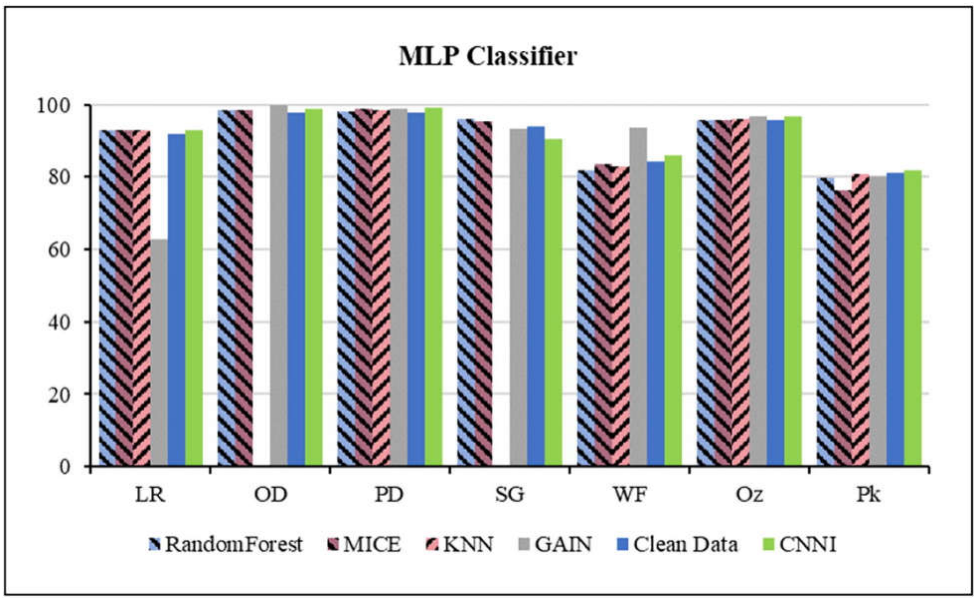


Рисунок 3.5 – Аналіз продуктивності методів імпутації для класифікатора MLP

На рисунках 3.1–3.5 вісь *x* показує набори даних, а вісь *y* – точність (%) класифікації.

Експериментальні результати, отримані за різних пропорцій пропущених значень (тобто 10%, 20%, 30%, 40%, 50%, 60%, 70%), наведені в таблиці 3.2 і візуалізовані на рисунках 3.6 та 3.7.

Таблиця 3.2 – Зміна співвідношення пропущених значень для досягнення стабільної точності

Набір даних	Класифікатор	10%	20%	30%	40%	50%	60%	70%
Optdigits	LR	96.902 ± 0.142	96.838 ± 0.222	95.998 ± 0.229	95.533 ± 0.330	95.338 ± 0.316	95.444 ± 0.294	95.373 ± 0.186
	LDA	95.755 ± 0.035	95.697 ± 0.267	94.933 ± 0.181	94.697 ± 0.090	94.644 ± 0.205	94.519 ± 0.261	94.608 ± 0.191
	KNN	98.843 ± 0.000	98.843 ± 0.000	98.833 ± 0.020	98.803 ± 0.100	98.841 ± 0.003	98.811 ± 0.000	98.801 ± 0.000
	MLP	98.624 ± 0.118	98.884 ± 0.354	98.669 ± 0.241	97.908 ± 0.342	97.758 ± 0.315	97.864 ± 0.368	97.918 ± 0.215
Waveform	LR	88.900 ± 0.451	88.920 ± 0.248	88.520 ± 0.519	87.020 ± 0.331	87.380 ± 0.318	87.300 ± 0.178	88.000 ± 0.792
	LDA	88.520 ± 0.097	88.440 ± 0.372	87.900 ± 0.360	87.520 ± 0.453	86.920 ± 0.530	86.840 ± 0.492	87.380 ± 0.430
	KNN	85.400 ± 0.000	85.421 ± 0.000	85.380 ± 0.000	85.400 ± 0.000	85.221 ± 0.000	85.400 ± 0.000	85.100 ± 0.000
	MLP	84.880 ± 0.722	85.860 ± 0.960	86.040 ± 0.233	85.700 ± 0.855	85.080 ± 0.563	85.500 ± 0.103	85.380 ± 0.847

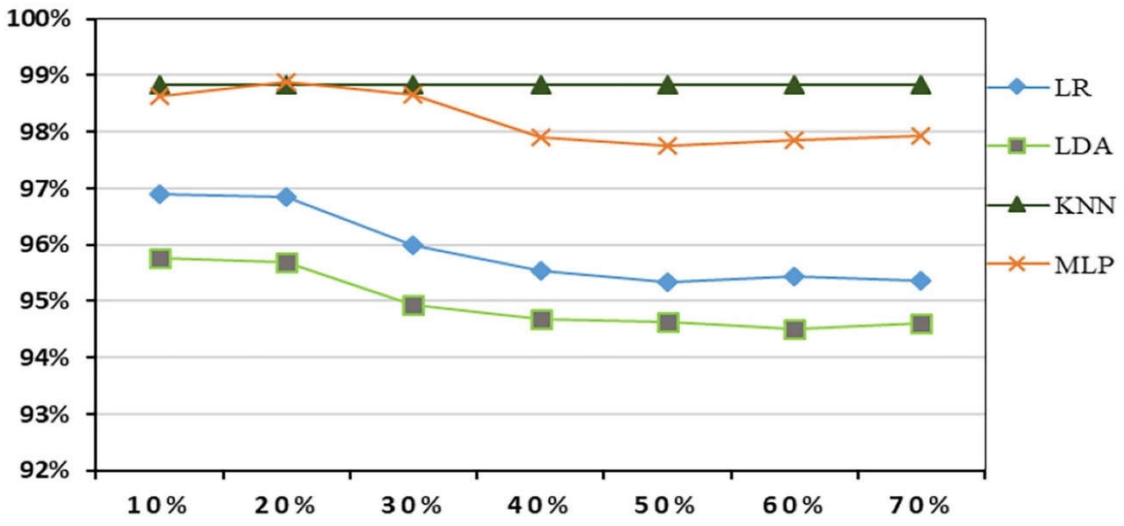


Рисунок 3.6 – Продуктивність при різних співвідношеннях пропущених значень на наборі даних Optdigits

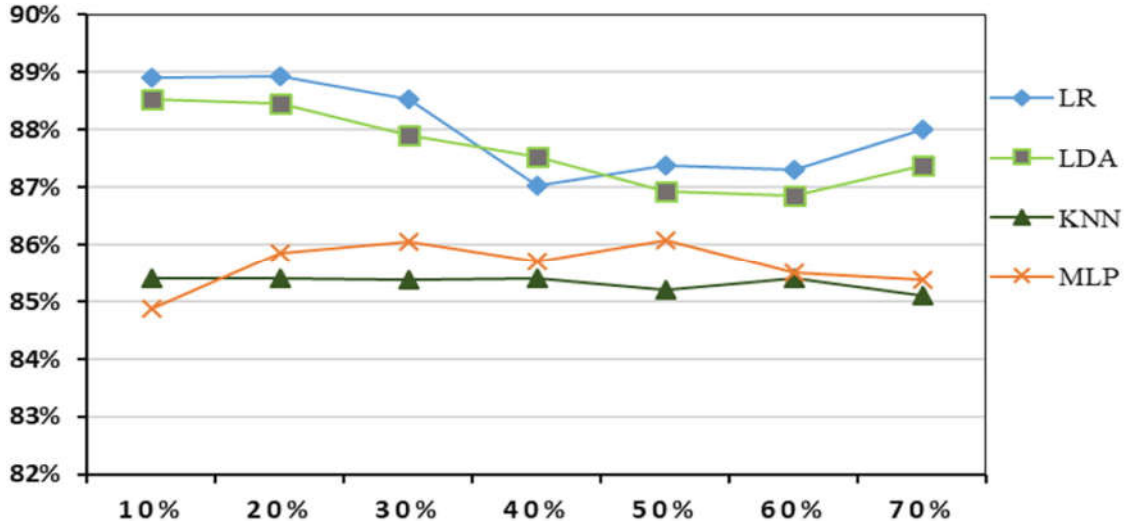


Рисунок 3.7 – Продуктивність при різних співвідношеннях пропущених значень на наборі даних Waveform

На рисунках 3.6 та 3.7 вісь *x* показує відсоткове співвідношення пропущених значень, а вісь *y* відображає точність класифікації (%).

Ці результати свідчать, що наша техніка імпутації демонструє стабільну продуктивність. Точність класифікації не змінюється суттєво при зміні пропорцій пропущених значень у наборах даних, що вказує на те, що зі збільшенням пропусків у наборі даних продуктивність запропонованого підходу не погіршиться очевидно.

Крім того, у таблиці 3.2 показано ефективність і доцільність використання запропонованого підходу при різних пропорціях пропущених значень. Результати, отримані на всіх наборах даних, показують, що імпутація, заснована на просторових взаємозв'язках у матриці даних, може призвести до покращення продуктивності класифікації.

Також запропонований підхід порівнюється з іншими передовими методами імпутації. Продуктивність запропонованого підходу порівнюється з чотирма іншими методами імпутації: GAIN, RandomForest, KNNI і MICE. У даних експериментах було використано загальнодоступний код із GitHub для реалізації підходу GAIN, а інші методи імпутації були реалізовані за допомогою бібліотеки Scikit-learn [31], де використовувалися налаштування параметрів за замовчуванням. Усі експерименти виконувалися на ПК з процесором Intel(R) Core(TM) i5-7400 CPU @ 3.00 GHz з 64-розрядною операційною системою.

Як видно з таблиці 3.2, точність класифікації, досягнута на даних, імпутованих за допомогою CNNI, краща, ніж на чистих даних, що свідчить про те, що імпутовані значення допомагають покращити представлення ознак за допомогою запропонованого підходу CNNI. Крім того, з експериментальних результатів видно, що у більшості випадків для всіх обраних класифікаторів продуктивність підходу CNNI є кращою, ніж інших методів. Однак результати показують, що для деяких класифікаторів на деяких наборах даних продуктивність даного методу трохи гірша. Наприклад, можна спостерігати, що середня точність класифікаторів KNN і MLP, навчених на наборах даних optdigits і waveform з імпутацією пропущених значень за допомогою методу GAIN, трохи вища, ніж у CNNI. Крім того, класифікатори NB показують кращу продуктивність на наборах даних pendigits і segmentation, використовуючи метод GAIN для імпутації пропущених значень. Random Forest показує кращу продуктивність на наборах даних pendigits і ozone, використовуючи класифікатори LR. Подібним чином, кращу продуктивність можна отримати на оригінальних (чистих) версіях наборів даних letter recognition і ozone, використовуючи класифікатори LR і NB відповідно, а метод KNNI показує кращу продуктивність на наборі даних letter recognition, використовуючи MLP для навчання класифікаторів.

Основна причина гарної продуктивності цих класифікаторів для деяких конкретних наборів даних, ймовірно, полягає у відповідності деяких конкретних алгоритмів навчання для наборів даних. Іншими словами, якщо набір даних добре відповідає структурі моделі класифікатора, то класифікатор покаже кращі результати.

Згідно з експериментальними результатами, даний метод демонструє значну перевагу, і в більшості випадків його продуктивність є або кращою, або дуже близькою до продуктивності методу GAIN. Підхід KNNI також показує хорошу продуктивність, але його обчислювальна ефективність низька через необхідність відстежувати всі навчальні записи для пошуку найближчих сусідів, тоді як CNNI може більш ефективно імпутувати дані, використовуючи згорткове ядро та мінімальну кількість параметрів.

Існують також певні обмеження запропонованого підходу, наприклад, у CNNI ваги ядра є параметрами, які визначаються шляхом навчання нелінійності, що існує в наданих даних, і оновлюються ітеративно. У такому випадку, якщо з даних не можна виявити просторові зв'язки, можливе неправильне навчання ядра, що призведе до неефективної імпутації. Оскільки робота зосереджена лише на числових наборах даних, але багато категоріальних наборів даних також можуть містити пропущені значення, це вказує на необхідність розглянути, як можна розширити запропонований підхід для обробки пропущених значень у категоріальних ознаках.

Експерименти з аналізу моделі було проведено для запропонованого підходу CNNI, і результати, отримані за допомогою різних налаштувань параметрів, наведені в таблиці 3.3. Проведено порівняння кожного конкретного налаштування параметрів із параметрами за замовчуванням і показано вплив налаштування параметрів на продуктивність. Тому, на основі результатів, показаних у таблиці 3.3, було обрано параметри за замовчуванням для запропонованого підходу CNNI, як зазначено вище.

Таблиця 3.3 – Аналіз моделі

Налаштування	LR	MLP
Налаштування за замовчуванням	96.838 ± 0.222	98.884 ± 0.354
CNN-DR →Plain 2D ConvNet	95.355 ± 0.344	90.782 ± 1.314
CNN-DR → Residual connection	95.362 ± 0.313	97.998 ± 0.255
CNN-DR → Dense connection	96.022 ± 0.451	98.352 ± 0.322
6 Conv. 2D layer→5 Conv. 2D layer	96.492 ± 0.745	98.659 ± 0.156
6 Conv. 2D layer→7 Conv. 2D layer	96.805 ± 0.603	98.885 ± 0.711
Epoch 500→300	95.409 ± 0.153	91.270 ± 0.699
Epoch 500→600	96.851 ± 0.337	96.679 ± 0.557
Kernel size 3×3→5×5	96.800 ± 0.153	98.199 ± 0.699
ReLU →Leaky ReLU	95.960 ± 0.306	98.590 ± 0.670
ReLU →Tanh	95.213 ± 0.220	87.651 ± 0.661
k = 3 →k = 4	96.851 ± 0.339	98.881 ± 0.318
k = 3 →k = 5	96.813 ± 0.121	98.879 ± 0.361

Аналіз моделі був проведений на наборі даних optdigits, і точність класифікації вимірювалася за допомогою двох алгоритмів навчання, а саме: багатошарового перцептрона (MLP) та логістичної регресії (LR) із різними налаштуваннями параметрів.

У таблиці 3.3 перший рядок представляє точність отриману за допомогою налаштувань параметрів за замовчуванням для CNNI на наборі даних optdigits з використанням класифікаторів LR і MLP. Як видно з таблиці 3.3, вплив налаштувань контрольованих експериментів можна оцінити для розуміння впливу різних елементів на згаданий набір даних.

Другий, третій і четвертий рядки таблиці 3.3 показують результати, отримані шляхом заміни архітектури за замовчуванням (CNN-DR) на просту 2D згорткову мережу (Conv. net), мережу з резидуальними з'єднаннями та щільну мережу CNN (Dense net). Зокрема, проста мережа складається з семи 2D згорткових шарів з активаційною функцією Leaky ReLU, у мережі з резидуальними з'єднаннями використовується з'єднання між вхідним та вихідним шарами, а в щільній мережі CNN кожен шар з'єднаний зі своїм наступником, а функцією активації є ReLU. Результати показують, що модель з параметрами за замовчуванням має кращу продуктивність порівняно з іншими параметрами.

П'ятий і шостий рядки таблиці 3.3 демонструють продуктивність при зміні кількості 2D згорткових шарів (Conv. layer). За замовчуванням використовується 6 шарів, але збільшення кількості шарів до 8 не призводить до значного покращення точності, що свідчить про те, що 6 шарів є достатніми для досягнення хорошої продуктивності.

Сьомий і восьмий рядки таблиці 3.10 показують вплив кількості епох на набір даних optdigits. У запропонованій моделі CNNI кількість епох за замовчуванням становить 500, але результати показують, що збільшення кількості епох не значно впливає на продуктивність, тоді як зменшення кількості епох має суттєвий вплив на результат.

Дев'ятий рядок таблиці 3.3 демонструє вплив розміру фільтра (ядра) на продуктивність. За замовчуванням обраний розмір ядра становить 3x3, але

експеримент показує, що збільшення розміру ядра погіршує продуктивність, що пов’язано з меншою впевненістю в імпутованих значеннях при збільшенні розміру ядра.

Останні два рядки таблиці 3.3 показують вплив кількості кластерів (k) на продуктивність. За замовчуванням було обрано k = 3, але різні значення k (4, 5) показали подібну точність, що свідчить про те, що метод не є надто чутливим до кількості кластерів. Однак збільшення кількості кластерів призводить до зростання обчислювальної складності, тому для експериментів обрано значення k = 3.

Висновки до розділу 3

1. Представлено набори даних із репозиторію UCI та налаштування параметрів для проведення експериментальних досліджень. Враховуючи відсутність початкових пропусків у більшості наборів даних, пропуски були створені штучно для оцінки ефективності імпутації. Запропонована архітектура CNN з шістьма 2D-згортковими шарами забезпечує стабільну продуктивність завдяки використанню оптимізатора Adam, функції активації ReLU та відповідних параметрів навчання, таких як нормалізація даних та правильний вибір кількості епох. Це дозволяє ефективно застосовувати підхід для імпутації пропущених значень у наборі даних.

2. Запропонований підхід був оцінений за допомогою точності класифікації для п’яти відомих алгоритмів навчання. Це дозволило визначити ефективність імпутації пропущених значень та її вплив на продуктивність класифікації. Результати підтвердили, що імпутовані дані можуть покращити якість навчання моделей порівняно з використанням лише чистих даних.

3. Експериментальні результати підтверджують, що запропонований підхід CNNI демонструє стабільну продуктивність у порівнянні з іншими методами імпутації пропущених значень. Навіть при різних співвідношеннях пропущених значень, продуктивність класифікації залишалася стабільною, що свідчить про надійність запропонованого підходу. Порівняння з іншими

сучасними методами, такими як GAIN, KNNI та MissForest, показало перевагу CNNI в більшості випадків, що підтверджує його ефективність для імпутації пропущених значень у числових наборах даних.

ВИСНОВКИ

У даній кваліфікаційній роботі досліджено підхід до імпутації пропущених даних на основі згорткових нейронних мереж, який спрямований на перетворення неповних даних на повні. У даному підході згорткова нейронна мережа створюється шляхом навчання просторових взаємозв'язків і нелінійності, що існують у наданих даних для імпутації пропущених значень. Крім того, для кластеризації даних використовується метод нечіткої кластеризації c-mean, що дозволяє організувати дані таким чином, щоб захопити сильну кореляцію між значеннями ознак.

Отримано наступні основні висновки:

1. Обробка пропущених даних є однією з ключових проблем у сучасній аналітиці, оскільки відсутність частини інформації може значно вплинути на точність і продуктивність моделей машинного навчання. Методи імпутації даних, такі як статистичні підходи, машинне навчання та глибинне навчання, мають свої переваги й обмеження залежно від структури і природи даних. Сучасні методи на основі глибокого навчання, зокрема згорткові нейронні мережі та генеративні змагальні мережі, демонструють великий потенціал для точного заповнення пропусків. Важливо враховувати просторові кореляції та складні зв’язки між ознаками, що дозволяє значно покращити якість даних і продуктивність класифікаційних моделей у різних сферах, таких як медицина, фінанси та соціальні науки.

2. Проведено аналіз основних підходів до імпутації пропущених значень, які можна розділити на статистичні методи та методи машинного навчання. Традиційні методи, такі як KNN і MICE, мають обмеження в продуктивності та потребують значних обчислювальних ресурсів, особливо для великих наборів даних. Методи, що базуються на глибинному навчанні, зокрема автоенкодери та генеративні змагальні мережі демонструють обнадійливі результати, але також потребують складної обчислювальної інфраструктури. У роботах підкреслюється важливість врахування просторових та нелінійних залежностей

між ознаками для точнішої імпутації, що є перспективним напрямом для розробки нових методів.

3. У сучасній аналітиці даних і машинному навчанні однією з найбільш поширених і критичних проблем є наявність пропущених значень у наборах даних. Пропуски в даних можуть суттєво вплинути на точність та ефективність моделей, що базуються на цих даних. Втрата частини інформації призводить до підвищеної похибки та зниження продуктивності моделей машинного навчання, особливо у випадках роботи з великими масивами даних, де просторові і нелінійні взаємозв'язки між ознаками є важливими. Головна задача цього дослідження полягає в тому, щоб запропонувати ефективний підхід для імпутації пропущених значень, який здатний працювати з великими наборами даних, забезпечуючи при цьому стабільну продуктивність навіть за умов значних пропусків.

4. Розглянуто метод імпутації пропущених значень на основі згорткових нейронних мереж і алгоритму нечіткої кластеризації FCM. Даний підхід використовує натреноване ядро CNN, яке виявляє просторові та нелінійні залежності між ознаками, для заповнення пропусків у даних. Алгоритм FCM допомагає ефективно організувати дані в кластери, покращуючи точність імпутації. Таким чином, цей підхід дозволяє покращити якість заповнення пропущених даних, що в свою чергу підвищує ефективність подальшого аналізу та класифікації.

5. Запропоновано архітектуру згорткової нейронної мережі, яка використовує щільні та резидуальні з'єднання для ефективної імпутації пропущених даних. Щільне з'єднання дозволяє кожному шару передавати інформацію всім наступним, забезпечуючи максимальне використання ознак. Резидуальні з'єднання допомагають вирішити проблему затухання градієнта, зберігаючи продуктивність глибокої нейронної мережі на високому рівні. Архітектура з шістьма згортковими шарами дозволяє ефективно заповнювати пропущені значення, що покращує якість даних та кінцеві результати моделі.

6. Розглянуто використання функції втрат у рамках даного підходу, який поєднує імпутацію пропущених даних та класифікацію. Основна мета функції

втрат полягає в підвищенні продуктивності класифікації, оскільки справжнє значення пропущених даних зазвичай невідоме. Використання категоріальної перехресної ентропії дозволяє порівнювати справжні та прогнозовані мітки, а зміна ваг згорткової мережі спрямована на мінімізацію цих втрат. Таким чином, спільне навчання моделі підвищує точність як імпутації пропущених значень, так і кінцевої класифікації.

7. Представлено набори даних із репозиторію UCI та налаштування параметрів для проведення експериментальних досліджень. Враховуючи відсутність початкових пропусків у більшості наборів даних, пропуски були створені штучно для оцінки ефективності імпутації. Запропонована архітектура CNN з шістьма 2D-згортковими шарами забезпечує стабільну продуктивність завдяки використанню оптимізатора Adam, функції активації ReLU та відповідних параметрів навчання, таких як нормалізація даних та правильний вибір кількості епох. Це дозволяє ефективно застосовувати підхід для імпутації пропущених значень у наборі даних.

8. Запропонований підхід був оцінений за допомогою точності класифікації для п’яти відомих алгоритмів навчання. Це дозволило визначити ефективність імпутації пропущених значень та її вплив на продуктивність класифікації. Результати підтвердили, що імпутовані дані можуть покращити якість навчання моделей порівняно з використанням лише чистих даних.

9. Експериментальні результати підтверджують, що запропонований підхід демонструє стабільну продуктивність у порівнянні з іншими методами імпутації пропущених значень. Навіть при різних співвідношеннях пропущених значень, продуктивність класифікації залишалася стабільною, що свідчить про надійність запропонованого підходу. Порівняння з іншими сучасними методами, такими як GAIN, KNNI та MissForest, показало перевагу запропонованого підходу в більшості випадків, що підтверджує його ефективність для імпутації пропущених значень у числових наборах даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Al-Milli N., Almobaideen W. Hybrid neural network to impute missing data for IoT applications. In: 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). IEEE. 2019. P. 121–125.

2. Andiojaya A., Demirhan H. A bagging algorithm for the imputation of missing values in time series. *Expert Systems with Applications.* 2019. Vol. 129. P. 10–26.

3. Bae B., Kim H., Lim H., Liu Y., Han L.D., Freeze P.B. Missing data imputation for traffic flow speed using spatio-temporal cokriging. *Transportation Research Part C: Emerging Technologies.* 2018. Vol. 88. P. 124–139.

4. Batista G.E., Monard M.C. An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence.* 2003. Vol. 17, No. 5–6. P. 519–533.

5. Beaulieu-Jones B.K., Moore J.H., P.R.O.-A.A.C.T. CONSORTIUM. Missing data imputation in the electronic health record using deeply learned autoencoders. In: *Pacific Symposium on Biocomputing 2017.* World Scientific. 2017. P. 207–218.

6. Belthangady C., Royer L.A. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nature Methods.* 2019. Vol. 16, No. 12. P. 1215–1225.

7. Beretta L., Santaniello A. Nearest neighbor imputation algorithms: a critical evaluation. *BMC Medical Informatics and Decision Making.* 2016. Vol. 16, No. 3. P. 197–208.

8. Bertsimas D., Pawlowski C., Zhuo Y.D. From predictive methods to missing data imputation: an optimization approach. *Journal of Machine Learning Research.* 2017. Vol. 18, No. 1. P. 7133–7171.

9. Che Z., Purushotham S., Cho K., Sontag D., Liu Y. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports.* 2018. Vol. 8, No. 1. P. 1–12.

10. Chen C.L., Mahjoubfar A., Tai L.C., Blaby I.K., Huang A., Niazi K.R., Jalali B. Deep learning in label-free cell classification. *Scientific Reports.* 2016. Vol. 6, No. 1. P. 1–16.

11. Chen X., Wei Z., Li Z., Liang J., Cai Y., Zhang B. Ensemble correlation-based low-rank matrix completion with applications to traffic data imputation. *Knowledge-Based Systems.* 2017. Vol. 132. P. 249–262.

12. Chen Y., Li Y., Narayan R., Subramanian A., Xie X. Gene expression inference with deep learning. *Bioinformatics.* 2016. Vol. 32, No. 12. P. 1832–1839.

13. Cheng C.-H., Chan C.-P., Sheu Y.-J. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. *Engineering Applications of Artificial Intelligence.* 2019. Vol. 81. P. 283–299.

14. Chiang H.-S., Chen M.-Y., Huang Y.-J. Wavelet-based EEG processing for epilepsy detection using fuzzy entropy and associative petri net. *IEEE Access.* 2019. Vol. 7. P. 103255–103262.

15. Choudhury S.J., Pal N.R. Imputation of missing data with neural networks for classification. *Knowledge-Based Systems.* 2019. Vol. 182. P. 104838.

16. De Jesús Rubio J., Lughofer E., Pieper J., Cruz P., Martinez D.I., Ochoa G., Islas M.A., Garcia E. Adapting H-infinity controller for the desired reference tracking of the sphere position in the maglev process. *Information Sciences.* 2021. Vol. 569. P. 669–686.

17. De Jesús Rubio J.J. Stability analysis of the modified Levenberg-Marquardt algorithm for the artificial neural network training. *IEEE Transactions on Neural Networks and Learning Systems.* Vol. 32, No. 8. P. 3510–3524.

18. De Souto M.C., Jaskowiak P.A., Costa I.G. Impact of missing data imputation methods on gene expression clustering and classification. *BMC Bioinformatics.* 2015. Vol. 16, No. 1. P. 1–9.

19. Dua D., Graff C. UCI Machine Learning Repository. URL: http://archive.ics.uci.edu/ml. 2017.

20. Duan Y., Lv Y., Kang W., Zhao Y. A deep learning based approach for traffic data imputation. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC).* IEEE. 2014. P. 912–917.

21. Garciarena U., Santana R. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications.* 2017. Vol. 89. P. 52–65.

22. Ge Z., Song Z., Ding S.X., Huang B. Data mining and analytics in the process industry: The role of machine learning. *IEEE Access.* 2017. Vol. 5. P. 20590–20616.

23. Gondara L., Wang K. Mida: Multiple imputation using denoising autoencoders. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Springer. 2018. P. 260–272.

24. Jaques N., Taylor S., Sano A., Picard R. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII).* IEEE. 2017. P. 202–208.

25. Khan H., Wang X., Liu H. Missing value imputation through shorter interval selection driven by Fuzzy C-Means clustering. *Computers and Electrical Engineering.* 2021. Vol. 93. P. 107230.

26. Kingma D.P., Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

27. Lai X., Wu X., Zhang L., Lu W., Zhong C. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing.* 2019. Vol. 366. P. 54–65.

28. Li S.C.-X., Jiang B., Marlin B. Misgan: Learning from incomplete data with generative adversarial networks. arXiv preprint arXiv:1902.09599.

29. López-González A., Campaña J.M., Martínez E.H., Contro P.P. Multi robot distance-based formation using Parallel Genetic Algorithm. *Applied Soft Computing.* 2020. Vol. 86. P. 105929.

30. McCombe N., Ding X., Prasad G., Finn D.P., Todd S., McClean P.L., Wong-Lin K. Predicting feature imputability in the absence of ground truth. arXiv preprint arXiv:2007.07052.

31. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A.,

Cournapeau D., Brucher M., Perrot M., Duchesnay E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research.* 2011. Vol. 12. P. 2825–2830.

32. Rahman M.G., Islam M.Z. Fimus: A framework for imputing missing values using co-appearance, correlation and similarity analysis. *Knowledge-Based Systems.* 2014. Vol. 56. P. 311–327.

33. Rubinsteyn A., Feldman S. fancyimpute: An Imputation Library for Python. URL: https://github.com/iskandr/fancyimpute. 2016.

34. Rubio J.J., Pan Y., Pieper J., Chen M.-Y., Sossa Azuela J.H. Advances in Robots Trajectories Learning via Fast Neural Networks. *Frontiers in Neurorobotics.* 2021. Vol. 15. P. 29.

35. Sefidian A.M., Daneshpour N. Missing value imputation using a novel grey-based fuzzy c-means, mutual information-based feature selection, and regression model. *Expert Systems with Applications.* 2019. Vol. 115. P. 68–94.

36. Silva-Ramírez E.-L., Pino-Mejías R., López-Coello M. Single imputation with multilayer perceptron and multiple imputation combining multilayer perceptron and k-nearest neighbours for monotone patterns. *Applied Soft Computing.* 2015. Vol. 29. P. 65–74.

37. Song S., Sun Y., Zhang A., Chen L., Wang J. Enriching data imputation under similarity rule constraints. *IEEE Transactions on Knowledge and Data Engineering.* 2018. Vol. 32, No. 2. P. 275–287.

38. Stekhoven D.J., Bühlmann P. MissForest – non-parametric missing value imputation for mixed-type data. *Bioinformatics.* 2012. Vol. 28, No. 1. P. 112–118.

39. Tang F., Ishwaran H. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal.* 2017. Vol. 10, No. 6. P. 363–377.

40. Tian J., Yu B., Yu D., Ma S. Missing data analyses: a hybrid multiple imputation algorithm using gray system theory and entropy based on clustering. *Applied Intelligence.* 2014. Vol. 40, No. 2. P. 376–388.

41. Tsai C.-F., Li M.-L., Lin W.-C. A class center-based approach for missing value imputation. *Knowledge-Based Systems.* 2018. Vol. 151. P. 124–135.

42. Van Hulse J., Khoshgoftaar T.M. Incomplete-case nearest neighbor imputation in software measurement data. *Information Sciences.* 2014. Vol. 259. P. 596–610.

43. Van Stein B., Kowalczyk W. An incremental algorithm for repairing training sets with missing values. In: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems.* Springer. 2016. P. 175–186.

44. Vargas D.M. Superpixels extraction by an intuitionistic fuzzy clustering algorithm. *Journal of Applied Research and Technology.* 2021. Vol. 19, No. 2. P. 140–152.

45. Webb S. Deep learning for biology. *Nature.* 2018. Vol. 554, No. 7693. P.555–557.

46. Xu X., Chong W., Li S., Arabo A., Xiao J. MIAEC: Missing data imputation based on the evidence chain. *IEEE Access.* 2018. Vol. 6. P. 12983–12992.

47. Yoon J., Jordon J., Schaar M. Gain: Missing data imputation using generative adversarial nets. In: *International Conference on Machine Learning.* PMLR. 2018. P. 5689–5698.

48. Zhang Q., Yuan Q., Zeng C., Li X., Wei Y. Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing.* 2018. Vol. 56, No. 8. P. 4274–4288.

49. Zhao L., Chen Z., Yang Z., Hu Y., Obaidat M.S. Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems. *IEEE Systems Journal.* 2016. Vol. 12, No. 2. P. 1610–1620.

50. Zhu B., Liu J.Z., Cauley S.F., Rosen B.R., Rosen M.S. Image reconstruction by domain-transform manifold learning. *Nature.* 2018. Vol. 555, No. 7697. P. 487–492.

51. Гончар Я. А., Єфанов Д. С. Сучасні підходи до управління проєктами: машинне навчання та блокчейн-технології. *Світ наукових досліджень* : матеріали міжнародної мультидисциплінарної наукової інтернет-конференції (випуск 35), м. Ополе, Польща, 20-21 листопада 2024 р. URL:

https://www.economy-confer.com.ua/full-article/5886/.

52. Гончар Я. А. Імпутація пропущених даних за допомогою глибоких нейронних мереж та нечіткої кластеризації. *Інформаційне суспільство: технологічні, економічні та технічні аспекти* становлення : матеріали міжнародної науково-практичної інтернет-конференції (випуск 94), м. Ополе, Польща, 11-12 грудня 2024 р. URL: http://www.konferenciaonline.org.ua/ua/article/id-1977/.

53. Островерхов В.М., Біловус Л.І., Возьний К.З., Луцишин О.О., Монастирський Г.Л., Надвиничний С.А., Питель С.В., Шандрук С.К. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

54. Комар М.П., Саченко А.О., Васильків Н.М., Загородня Д.І. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за другим (магістерським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 32 с.

Додаток А

Приклад реалізації згорткової нейронної мережі для імпутації пропущених даних

```
import tensorflow as tf
from tensorflow.keras import layers, models, optimizers

# Функція для створення архітектури CNN
def create_cnn_model(input_shape):
    model = models.Sequential()

    # Перший згортковий шар з 16 ядрами 3x3
    model.add(layers.Conv2D(16, (3, 3), padding='same', activation='relu', input_shape=input_shape))

    # Другий згортковий шар з 32 ядрами 3x3
    model.add(layers.Conv2D(32, (3, 3), padding='same', activation='relu'))

    # Третій згортковий шар з 64 ядрами 3x3
    model.add(layers.Conv2D(64, (3, 3), padding='same', activation='relu'))
```

```
# Четвертий згортковий шар з 128 ядрами 3x3
model.add(layers.Conv2D(128, (3, 3), padding='same', activation='relu'))

# П'ятий згортковий шар з 256 ядрами 3x3
model.add(layers.Conv2D(256, (3, 3), padding='same', activation='relu'))

# Flatten шар для перетворення даних в одномірний вектор
model.add(layers.Flatten())

# Повнозв'язний шар
model.add(layers.Dense(512, activation='relu'))

# Вихідний шар з функцією softmax для класифікації
model.add(layers.Dense(10, activation='softmax'))  # Для класифікації 10 класів (залежить від задачі)

# Компіляція моделі з використанням оптимізатора Adam і функції втрат categorical_crossentropy
model.compile(optimizer=optimizers.Adam(learning_rate=1e-3),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

```
    return model


# Приклад використання:
input_shape = (64, 64, 1)  # Розмір вхідних даних (залежить від задачі, наприклад, зображення 64x64)
model = create_cnn_model(input_shape)


# Виведення архітектури моделі
model.summary()


# Навчання моделі
# X_train і y_train — навчальні дані та мітки (їх треба підготувати заздалегідь)
# model.fit(X_train, y_train, epochs=500, batch_size=4)
```

Додаток Б

Точність класифікації, отримана на різних наборах даних

Таблиця Б.1 – Точність класифікації, отримана на наборі даних Pendigits

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	92.871 ± 0.077	91.915 ± 0.469	92.700 ± 0.469	90.807 ± 0.550	91.859 ± 0.000	88.331 ± 0.307
LDA	86.606 ± 0.027	87.174 ± 0.593	87.413 ± 0.703	87.234 ± 0.609	86.857 ± 0.000	87.788 ± 0.262
KNN	99.488 ± 0.000	99.124 ± 0.253	99.215 ± 0.222	99.359 ± 0.205	98.908 ± 0.000	99.617 ± 0.018
NB	85.730 ± 0.000	86.344 ± 0.821	86.048 ± 0.848	86.431 ± 0.915	85.629 ± 0.000	81.473 ± 0.068
MLP	98.215 ± 0.151	98.811 ± 0.105	98.635 ± 0.450	98.975 ± 0.296	97.772 ± 0.000	98.988 ± 0.231

Таблиця Б.2 – Точність класифікації, отримана на наборі даних Optdigits

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	96.450 ± 0.084	96.733 ± 0.388	—	95.647 ± 1.923	96.797 ± 0.000	96.838 ± 0.222
LDA	95.063 ± 0.055	94.933 ± 1.138	—	93.867 ± 3.002	95.618 ± 0.000	95.697 ± 0.267
KNN	98.019 ± 0.042	98.622 ± 0.499	—	99.489 ± 0.625	98.833 ± 0.000	98.843 ± 0.000
NB	91.290 ± 0.162	91.666 ± 0.810	—	87.815 ± 4.805	91.459 ± 0.000	91.879 ± 0.297
MLP	98.489 ± 0.116	98.400 ± 0.268	—	100.00 ± 0.000	97.754 ± 0.000	98.884 ± 0.354

Таблиця Б.3 – Точність класифікації, отримана на наборі даних Letter Recognition

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	71.085 ± 0.040	71.287 ± 0.684	71.206 ± 0.787	41.205 ± 0.870	70.825 ± 0.000	71.465 ± 0.302
LDA	69.708 ± 0.014	69.556 ± 0.825	69.625 ± 0.970	41.232 ± 0.735	69.680 ± 0.000	69.975 ± 0.188
KNN	94.684 ± 0.104	94.693 ± 0.825	94.631 ± 0.253	76.571 ± 1.474	94.225 ± 0.000	94.930 ± 0.131
NB	63.246 ± 0.024	63.081 ± 0.465	63.250 ± 0.455	38.345 ± 0.919	64.025 ± 0.000	63.730 ± 0.452
MLP	92.882 ± 0.311	92.950 ± 0.341	92.987 ± 0.275	62.923 ± 1.417	92.025 ± 0.000	92.850 ± 0.369

Таблиця Б.4 – Точність класифікації, отримана на наборі даних Segment

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	92.571 ± 0.246	92.417 ± 1.363	—	92.566 ± 1.518	93.285 ± 0.000	94.022 ± 0.407
LDA	92.046 ± 0.060	90.494 ± 0.992	—	92.372 ± 1.298	90.769 ± 0.000	92.406 ± 0.291
KNN	94.274 ± 0.237	94.120 ± 0.788	—	96.378 ± 1.326	92.747 ± 0.000	94.659 ± 0.107
NB	81.318 ± 0.000	79.230 ± 2.820	—	81.988 ± 2.581	79.560 ± 0.000	70.769 ± 2.103
MLP	96.035 ± 0.759	95.439 ± 0.788	—	93.257 ± 1.579	94.065 ± 0.000	90.461 ± 2.017

Таблиця Б.5 – Точність класифікації, отримана на наборі даних Parkinson

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	61.930 ± 0.084	59.489 ± 0.918	62.383 ± 1.578	60.471 ± 1.729	52.068 ± 0.000	62.978 ± 0.789
LDA	80.580 ± 0.063	74.680 ± 1.095	79.978 ± 1.533	70.222 ± 1.850	81.191 ± 0.000	70.127 ± 0.559
KNN	69.481 ± 0.169	68.638 ± 0.827	68.638 ± 0.950	81.947 ± 1.038	69.702 ± 0.000	69.868 ± 0.393
NB	51.144 ± 0.045	43.212 ± 0.518	48.255 ± 1.266	58.945 ± 2.466	53.531 ± 0.000	28.187 ± 0.549
MLP	79.639 ± 0.217	76.170 ± 1.926	80.808 ± 1.071	79.957 ± 0.993	81.276 ± 0.000	81.863 ± 1.623

Таблиця Б.6 – Точність класифікації, отримана на наборі даних Waveform

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	86.595 ± 0.090	86.925 ± 0.430	86.325 ± 1.406	86.120 ± 0.577	88.400 ± 0.000	88.920 ± 0.248
LDA	86.018 ± 0.070	86.350 ± 0.532	85.925 ± 1.130	85.780 ± 0.737	88.300 ± 0.000	88.440 ± 0.372
KNN	81.668 ± 0.144	82.375 ± 0.079	81.850 ± 1.441	87.689 ± 0.671	85.400 ± 0.000	85.421 ± 0.000
NB	81.224 ± 0.050	81.250 ± 0.684	80.000 ± 1.015	79.690 ± 1.073	81.500 ± 0.000	81.851 ± 0.080
MLP	81.909 ± 0.746	83.550 ± 0.422	82.875 ± 0.932	93.760 ± 1.475	84.400 ± 0.000	85.860 ± 0.960

Таблиця Б.7 – Точність класифікації, отримана на наборі даних Ozone

Класифікатор	RandomForest	MICE	KNNI	GAIN	Clean data	CNNI
LR	96.538 ± 0.145	96.418 ± 0.344	96.213 ± 0.311	95.891 ± 2.537	94.594 ± 0.000	94.756 ± 0.366
LDA	96.283 ± 0.000	96.148 ± 0.165	96.451 ± 0.211	95.891 ± 2.537	94.594 ± 0.000	96.943 ± 0.216
KNN	96.113 ± 0.000	96.283 ± 0.302	96.666 ± 0.006	95.891 ± 2.537	95.405 ± 0.000	96.880 ± 0.000
NB	71.959 ± 0.000	70.270 ± 1.282	71.556 ± 0.052	72.100 ± 9.997	77.567 ± 0.000	72.621 ± 0.556
MLP	95.578 ± 0.469	95.743 ± 0.727	95.986 ± 0.631	96.761 ± 3.008	95.675 ± 0.000	96.897 ± 0.132

Додаток В

Копії публікацій

У веб-додаток при потребі ще можуть бути внесені деякі корективи, але тепер основний напрямок роботи з словниками – це їх розширення, збільшення в них кількості слів та виразів.

Список літератури:

1. Bootstrap 5 підручник. – URL: https://w3schoolsua.github.io/bootstrap/bootstrap_ver.html
2. PHP підручник – URL: https://w3schoolsua.github.io/php/index.html#gsc.tab=0
3. MySQL 8.0 Release Notes – URL: https://dev.mysql.com/doc/relnotes/mysql/8.0/en/
4. ResponsiveVoice Text to Speech API – URL: https://responsivevoice.org/api/
5. Voice is a keyboard – URL: https://voicehotkey.com/

МЕТОДИ ВІДНОВЛЕННЯ ТА ОЦІНКИ ЯКОСТІ ВЕЛИКИХ ДАНИХ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ

Гончар Ярослав Андрійович
магістрант, Західноукраїнський національний університет, м. Тернопіль

Єфанов Дмитро Сергійович
магістрант, Західноукраїнський національний університет, м. Тернопіль

Інтернет-адреса публікації на сайті:
https://www.economy-confer.com.ua/full-article/5886/

У сучасному цифровому світі великі дані стали основою для прийняття рішень у різних галузях, таких як медицина, фінанси, транспорт та промисловість. Проте значна частина даних часто є неповною, пошкодженою або має низьку якість через технічні збої, людські помилки або обмеження сенсорних пристроїв. Відсутність або низька якість даних може суттєво впливати на точність моделей аналізу і прогнозування, що ускладнює їх використання в критично важливих задачах.

Нейронні мережі та методи глибокого навчання відкривають нові можливості для автоматизованого відновлення даних, виявлення аномалій та оцінки якості великих даних. Вони дозволяють виявляти приховані закономірності, відновлювати втрачені значення та забезпечувати комплексну оцінку даних з високим рівнем адаптивності [1, 2].

Одна з поширених проблем у роботі з даними – це наявність пропущених значень, що може створити проблеми під час аналізу. Такі пропущені дані трапляються в різних сферах, таких як аналіз експресії генів, контроль дорожнього руху, промислові інформаційні системи, обробка зображень і розробка програмного забезпечення. Якщо цю проблему не врахувати під час аналізу, це може призвести до неправильних висновків і результатів. Тому

важливо підвищити якість даних, обробляючи пропущені значення належним чином.

Існують два основні традиційні методи обробки пропущених даних. Перший метод полягає в тому, щоб просто видалити всі записи, де відсутні якісь дані. Другий метод полягає в тому, щоб замінити пропущені значення на якісь припустимі значення. Крім того, існують методи імпутації (заміщення) даних, які використовують машинне навчання. Наприклад, до таких методів належать метод найближчих сусідів (KNN), рекурентні нейронні мережі (RNN), і генеративні змагальні мережі для імпутації даних (GAIN) [3-5].

За останні кілька років глибоке навчання стало широко використовуватися для вирішення різних задач, включаючи заміщення пропущених даних. Використання великих обсягів навчальних даних дозволило значно покращити результати імпутації. Зокрема, генеративні змагальні мережі (GAN) показали великі успіхи у вирішенні цієї задачі. Наприклад, один із методів імпутації на основі GAN вимагав налаштування гіперпараметрів для регулювання функції втрат і швидкості роботи дискримінатора. В іншому підході GAN використовував генератор і дискримінатор окремо для навчання структури і розподілу пропущених даних. Хоча ці методи дають чудові результати, вони часто занадто складні для практичного використання через велику кількість налаштувань [1, 2].

Для отримання достовірних і якісних результатів при використанні інформаційних технологій важливе значення мають не лише методи, способи та засоби обробки даних, але й якість самих вихідних даних. Від таких характеристик, як повнота, точність і змістовність, безпосередньо залежить результативність застосованих технологій. Деякі з цих характеристик можуть мати більшу вагу в конкретному контексті, але разом вони формують основу для оцінки якості результатів, отриманих із цих даних.

Проте, у процесі використання інформаційних технологій часто виникають проблеми, пов’язані з наявністю неповних або надлишкових даних. Такі ситуації потребують оцінки якості початкових даних, оскільки вони безпосередньо впливають на кінцевий результат. Сучасні технології обробки даних зазвичай працюють із великими обсягами різнотипних даних, які, хоча й численні, можуть не відповідати вимогам якості.

Особливо це актуально для глибокого навчання, яке використовує штучні нейронні мережі. Ці моделі вимагають великих і якісних наборів даних для формування потужних абстракцій. Однак навіть у сценаріях із великими даними їх якість може бути недостатньою для ефективного навчання. Невеликі варіації, несподівані особливості чи неповнота початкових даних здатні суттєво порушити баланс у навчальних моделях нейронних мереж, що негативно впливає на їхню точність і стабільність.

Враховуючи ці виклики, необхідність початкової оцінки якості великих даних є критичною. Зокрема, це особливо важливо для інформаційних технологій, побудованих на сучасних методах, таких як інтелектуальні та еволюційні алгоритми. Оцінка якості даних дозволяє забезпечити їх

відповідність вимогам і знизити ризик виникнення помилок у процесі аналізу та прогнозування.

Отже, методи відновлення та оцінки якості великих даних на основі нейронних мереж дозволять забезпечити високу точність і адаптивність у роботі з неповними та низькоякісними даними, що є критично важливим для сучасних інформаційних технологій. Використання глибокого навчання, зокрема автоенкодерів та генеративних змагальних мереж, дозволить ефективно заповнювати пропущені дані та оцінювати їх якість, мінімізуючи ризик помилок у моделюванні. Ці підходи відкривають нові можливості для підвищення надійності аналізу даних у різних галузях.

Література:

1. Choudhury S. J., Pal N. R. Imputation of missing data with neural networks for classification. Knowledge-Based System. 2019. 182. 104838.
2. Lai, X., Wu, X., Zhang, L., Lu, W., Zhong, C. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. Neurocomputing. 2019. 366. 54-65.
3. Bertsimas, D., Pawlowski, C., Zhuo, Y.D. From predictive methods to missing data imputation: an optimization approach. J. Mach. Learn. Res. 2017. 18(1). 7133-7171.
4. Cheng, C.-H., Chan, C.-P., Sheu, Y.-J. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. Eng. Appl. Artif. Intell. 2019. 81. 283-299.
5. Tang, F., Ishwaran, H. Random forest missing data algorithms. Statistical Analysis and Data Mining: The ASA Data Sci. J. 2017. 10(6). 363-377.

АНАЛІЗ ТА ДОСЛІДЖЕННЯ СИСТЕМИ АВТОМОБІЛЬ-ЗОВНІШНЄ СЕРЕДОВИЩЕ

Горб Ніктіа Станіславович
студент кафедри комп'ютеризованих систем автоматики, Національний університет «Львівська політехніка»

Науковий керівник: Наконечний Адріан Йосифович
доктор технічних наук, Національний університет «Львівська політехніка»

Інтернет-адреса публікації на сайті:
https://www.economy-confer.com.ua/full-article/5837/

Зростаюча кількість транспортних засобів і підвищені вимоги до безпеки дорожнього руху стимулюють розвиток технологій, які дозволяють транспортним системам взаємодіяти з дорожньою інфраструктурою. Інтеграція систем автомобіль-зовнішнє середовище відкриває нові перспективи для оптимізації дорожнього руху, зниження аварійності та підвищення загальної ефективності транспортних потоків. У роботі досліджено можливості

Вас вітає Інтернет конференція!

Вітаємо на нашому сайті

Рік заснування видання - 2011

ІМПУТАЦІЯ ПРОПУЩЕНИХ ДАНИХ ЗА ДОПОМОГОЮ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ ТА НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ

01.12.2024 13:03

[1. Інформаційні системи і технології]

Автор: **Гончар Ярослав Андрійович, магістр, Західноукраїнський національний університет, м. Тернопіль**

Пропущені дані є поширеним явищем у будь-якій реальній базі даних або інформаційній системі. Вони виникають через різні причини, такі як людські помилки, технічні збої, неспроможність отримати певну інформацію або свідоме рішення не включати певні дані. Пропуски можуть бути випадковими (MCAR - Missing Completely at Random), залежати від спостережуваних даних (MAR - Missing at Random) або залежати від неспостережуваних чинників (MNAR - Missing Not at Random) [1]. Від того, як виникли пропуски, залежить ефективність різних методів їх імпутації.

Сучасні методи імпутації пропущених даних можна розділити на три основні категорії: статистичні методи, методи машинного навчання [2-4] та підходи, засновані на глибокому навчанні [5, 6]. Кожен з цих методів має свої переваги та недоліки, і вибір конкретного підходу залежить від структури даних, обсягу пропусків та обчислювальних можливостей.

Основними викликами є ефективна обробка пропущених даних у великих наборах даних, де існують складні взаємозв'язки між ознаками. Методи, які враховують просторові кореляції, такі як згорткові нейронні мережі, мають значний потенціал для підвищення точності імпутації.

Глибокі нейронні мережі (ГНМ) є перспективним вибором для задач імпутації даних завдяки їхнім унікальним можливостям у роботі з великими та складними наборами даних. ГНМ здатні автоматично виявляти та моделювати нелінійні залежності між ознаками. Це особливо важливо для даних із високим рівнем складності, де прості моделі, як-от статистичні або лінійні, виявляються недостатньо точними. Завдяки своїй багаторівневій архітектурі ГНМ можуть ефективно працювати з великими наборами даних, забезпечуючи високу точність навіть у разі складних структур даних або великої кількості змінних. ГНМ можуть працювати зі структурованими, напівструктурованими та неструктурованими даними, такими як числові дані, текст або зображення. Це універсальність робить їх придатними для широкого спектра застосувань.

Архітектури, такі як згорткові нейронні мережі (ЗНМ) та рекурентні нейронні мережі, дозволяють враховувати просторові або часові залежності, що є важливими для багатьох реальних задач, включаючи аналіз часових рядів або обробку зображень. ГНМ не потребують ручного проектування ознак, оскільки вони автоматично виділяють найважливіші патерни з даних. Це спрощує підготовку даних і покращує якість результатів. Завдяки своїй архітектурі ГНМ можуть навчатися навіть на даних із пропущеними значеннями або шумом, відновлюючи відсутню інформацію з високою точністю. ГНМ підтримують широкий спектр налаштувань і архітектур, які можна адаптувати до специфічних задач, включаючи імпутацію даних, прогнозування або класифікацію.

Вибір ГНМ для задач імпутації пропущених значень обґрунтовується їхньою здатністю працювати з великими обсягами даних, моделювати складні залежності та забезпечувати високий рівень точності результатів.

Запропонований метод відновлення пропущених даних базується на поєднанні алгоритму нечіткої кластеризації Fuzzy C-Means (FCM) і ЗНМ. Основна ідея методу полягає у використанні просторових залежностей і нелінійних закономірностей, щоб точно заповнити відсутні значення в даних.

На першому етапі метод аналізує кореляції між ознаками, щоб найбільш взаємопов’язані ознаки були розташовані поруч. Це дозволяє ефективніше використовувати інформацію про зв’язки між даними під час навчання моделі. Потім застосовується алгоритм FCM, який організовує дані у кілька кластерів. Особливість цього алгоритму полягає в тому, що кожен запис може належати одночасно до кількох кластерів із різними ступенями членства. Таке впорядкування дозволяє класифікувати записи за їхньою схожістю та підготувати їх для подальшої обробки.

Після кластеризації дані передаються до згорткової нейронної мережі. Модель ЗНМ використовує навчуване ядро, щоб заповнити пропуски у даних. Ядро аналізує просторові взаємозв’язки між значеннями у сусідніх записах і визначає відсутні значення шляхом згортки. Процес навчання моделі включає оптимізацію ваг ядра ЗНМ, що дозволяє мінімізувати помилки заповнення.

Під час тестування модель ЗНМ заповнює пропущені значення у нових даних. Ці значення виділяються через маскування й об’єднуються з уже наявними даними, утворюючи повний набір. Завдяки цьому підходу модель може точно заповнювати пропуски навіть у складних наборах даних із нелінійними зв’язками.

Основними перевагами методу є його здатність працювати з великими та різнорідними даними, використання інформації про кореляції й просторові зв’язки, а також висока точність імпутації. Цей підхід особливо ефективний у сферах, де важливо зберігати приховані закономірності в даних, таких як аналіз статистичних даних, обробка зображень чи машинне навчання.

Література

1. He, Y. Missing data analysis using multiple imputation. Circulation: Cardiovascular Quality and Outcomes. 2010. Vol. 3(1). Pp. 98–105.

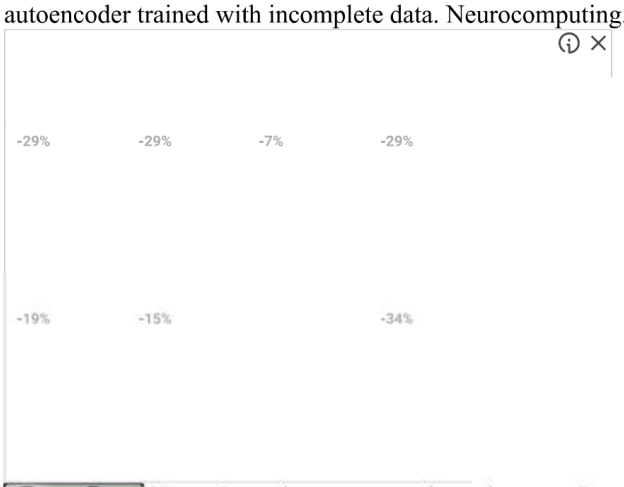
2. Beretta, L., Santaniello, A. Nearest neighbor imputation algorithms: a critical evaluation. BMC Med. Inform. Decision Making. 2016. Vol. 16(3). Pp. 197–208.

3. Cheng, C.-H., Chan, C.-P., Sheu, Y.-J. A novel purity-based k nearest neighbors imputation method and its application in financial distress prediction. Eng. Appl. Artif. Intell. 2019. Vol. 81. Pp.283–299.

4. Tang, F., Ishwaran, H. Random forest missing data algorithms. Statistical Analysis and Data Mining: The ASA Data Sci. J. 2017. Vol. 10(6). Pp.363–377.

5. Choudhury, S.J., Pal, N.R. Imputation of missing data with neural networks for classification. Knowledge-Based Syst. 2019. Vol. 182. №104838.

6. Lai, X., Wu, X., Zhang, L., Lu, W., Zhong, C. Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. Neurocomputing. 2019. Vol. 366. Pp.54–65.



Інші наукові праці даної секції

- СИСТЕМА КОНТРОЛЮ ПОКАЗНИКІВ СЕРЕДОВИЩА
  30.11.2024 20:38
- УПРАВЛІННЯ ПРОЄКТАМИ ТА ВПРОВАДЖЕННЯ СИСТЕМИ ДЛЯ ЕФЕКТИВНОГО МОНІТОРИНГУ І УПРАВЛІННЯ РЕЗЕРВАМИ НА ОСНОВІ БЛОКЧЕЙН ТЕХНОЛОГІЇ
  01.12.2024 13:54
- ПРОГНОЗУВАННЯ ФІНАНСОВИХ ПОКАЗНИКІВ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ
  01.12.2024 13:51
- УПРАВЛІННЯ ПРОЄКТАМИ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ ТА НЕЙРОННИХ МЕРЕЖ
  01.12.2024 13:48
- УПРАВЛІННЯ СКЛАДСЬКИМИ ЗАПАСАМИ НА ОСНОВІ ІНТЕГРАЦІЇ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ ТА МАШИННОГО НАВЧАННЯ
  01.12.2024 13:46

Конференції

Конференції 2024

Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 84) (18-19.01.2024)

- 1. Інформаційні системи і технології 28
- 2. Економічні науки 14
- 3. Технічні науки 12

Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 85) (15-16.02.2024)