

Міністерство освіти і науки України
Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота

за освітнім ступенем «магістр» на тему:

**КЛАСИФІКАЦІЯ ЗАДАЧ ДЛЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ
РІВНЯНЬ МЕТОДАМИ МАШИННОГО НАВЧАННЯ**

Виконав:

здобувач 2 курсу групи М-КН-21

спеціальності 122 «Комп'ютерні науки»

Мороз Ігор Петрович

Науковий керівник:

кандидат технічних наук, доцент

Сяський Володимир Андрійович

РЕФЕРАТ

Розглянуто проблему автоматизації процесу розв'язання задач математичної фізики, які характеризуються складністю, методами теорії збурень асимптотичного аналізу. Для вирішення проблеми пропонується залучити методи глибокого машинного навчання.

Об'єкт дослідження – методи і технології інтелектуального аналізу.

Предметом дослідження є вивчення можливості використання нейронних мереж для автоматизованої класифікації звичайних диференціальних рівнянь (ЗДР) і відповідних постановок задач.

У роботі запропоновано концепцію розробки штучних нейронних мереж ідентифікації задач Коші (крайових задач) для звичайних диференціальних рівнянь, які з'являються у результаті розчеплення деякої базової задачі математичної фізики. Розглянуто відповідну джерельну базу, методи класифікації (ідентифікації) об'єктів, типи штучних нейронних мереж, інструментальні засоби їх реалізації.

Кваліфікаційна робота складається зі змісту, вступу, трьох розділів, висновків та списку використаних літературних джерел із 33 найменувань. До роботи додається реферат українською та англійською мовами. Загальний обсяг роботи складає 55 сторінок.

Ключові слова: розпізнавання образів, інтелектуальний аналіз, класифікація, звичайні диференціальні рівняння.

ABSTRACT

The problem automation of the process of solving problems of mathematical physics, which are characterized by complexity, by methods of the asymptotic analysis perturbation theory is considered. To solve the problem, it is proposed to use methods of deep machine learning.

The **object** of research is methods and technologies of the intellectual analysis.

The **subject** of the research is to study the possibility of using artificial neural networks for automated classification of ordinary differential equations (ODEs) and corresponding problem statements.

The paper proposes a concept for developing artificial neural networks for identifying Cauchy problems (boundary value problems) for ordinary differential equations that arise as a result of decoupling some basic problem of mathematical physics. The corresponding source base, methods of classifying (identifying) objects, types of artificial neural networks, and tools for their implementation are considered.

The qualification work consists of table of contents, introduction, three chapters, conclusions and a list of 33 references. An abstract in Ukrainian and English is attached to the work. The total contents of work are 55 pages.

Key words: pattern recognition, machine learning, classification, ordinary differential equations.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ. ПОТОЧНИЙ СТАН І ПЕРСПЕКТИВИ РОЗВИТКУ	7
1.1. Інформаційні системи на робочому місці математика	7
1.2. Математичне моделювання складних систем. Засоби і шляхи підвищення ефективності моделювання	10
1.2.1. Асимптотичні методи в моделюванні складних систем.....	10
1.2.2. Алгоритм розв'язання сингулярно збурених задач напівпровідникової електроніки методом теорії збурень.....	11
1.2.3. Проблема автоматизації процесу розв'язання задач асимптотичними методами.....	19
РОЗДІЛ 2. КЛАСИФІКАЦІЯ ТА ІДЕНТИФІКАЦІЯ ЗАДАЧ ДЛЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ.....	21
2.1. Подання вхідних даних. Опис системи ознак.....	21
2.2. Постановка задач класифікації та ідентифікації.....	25
2.3. Методи і моделі машинного навчання.....	28
2.4. Метрики якості в задачах класифікації.....	35
РОЗДІЛ 3. НЕЙРОМЕРЕЖЕВІ МОДЕЛІ В ЗАДАЧАХ КЛАСИФІКАЦІ ТА ІДЕНТИФІКАЦІ.....	37
3.1. Нейронні мережі. Базові поняття.....	37
3.2. Моделі нейронних мереж (архітектура НМ).....	40
3.3. Інструментальні засоби реалізації	43
3.4. Синтез нейронних мереж для вирішення задачі.....	45
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	49
ДОДАТКИ.....	53
Додаток А. Сертифікат учасника конференції.....	53

ВСТУП

Проблема підвищення ефективності обчислювальних алгоритмів розв'язання різноманітних модельних задач (наукових, інженерних, економічних тощо) постійно привертає увагу дослідників. Обумовлено це як і зростанням рівня складності задач через удосконалення відповідних математичних моделей, врахуванням нових факторів впливу на результат, нових механізмів взаємодій, їх комбінуванням тощо, так і удосконаленням обчислювальних систем, появою нових інструментальних засобів і інформаційних технологій. За умов ускладнення практичних задач, що вирішуються у наш час, важливими питаннями є забезпечення результативності алгоритмів, досягнення необхідної точності обчислень, мінімізації часу виконання алгоритмів, досягнення оптимальних показників використаних ресурсів обчислювальної системи.

Високу ефективність розв'язання інженерних задач, які зазвичай подаються у вигляді крайових задач для систем диференціальних чи (або) інтегро-диференціальних рівнянь, демонструють асимптотичні методи, зокрема методи теорії збурень [1,2]. Наприклад, для розв'язання типових нелінійних сингулярно збурених задач напівпровідникової електроніки використовують метод примежових поправок [3,4], який дозволяє отримати інформацію про електрофізичні характеристики відповідних електронних пристроїв.

Асимптотичні методи забезпечують приведення складної початкової задачі до рекурентної послідовності задач нижчого рівня складності, які у деяких випадках розв'язуються аналітичними (точними) методами. Відповідні алгоритми розчеплення (декомпозиції) початкової задачі реалізуються символьними перетвореннями і такий підхід забезпечує збереження визначеного рівня точності результату без збільшення часу виконання відповідних алгоритмів.

Проте вирішенню проблеми автоматизації процедури розчеплення задач заважає необхідність визначення типів задач, які формуються на певних етапах рекурентних перетворень, для передачі їх відповідним процедурам символьного

або чисельного розв'язання. Для усунення цієї перепони виникла ідея застосування елементів інтелектуального аналізу з використанням відповідних методів і технологій.

Метою роботи є розробка концепцій нейронних мереж з різними алгоритмами машинного навчання класифікації задач для звичайних диференціальних рівнянь.

Для досягнення мети дослідження визначені наступні **завдання**:

- дослідити предметну область (методологію розв'язання сингулярно збурених модельних задач асимптотичними методами; структуру математичних постановок типових задач, отриманих у результаті проведення декомпозиції);
- дослідити методи класифікації образів;
- здійснити постановку задачі класифікації ЗДР;
- розробити процедуру класифікації постановок задач для ЗДР;
- дослідити інструментальні засоби інтелектуального аналізу для реалізації відповідної процедури класифікації;
- запропонувати відповідні концептуальні рішення.

Об'єкт дослідження – методи і технології інтелектуального аналізу.

Предметом дослідження є вивчення можливості використання нейронних мереж для автоматизованої класифікації звичайних диференціальних рівнянь (ЗДР) і відповідних постановок задач.

Методи досліджень. Застосовано загальнонаукові підходи до пошуку і системного аналізу інформації за тематикою дослідження. Використано метод примежових поправок для проведення декомпозиції нелінійної сингулярно збуреної задачі напівпровідникової електроніки, методи класифікації теорії розпізнавання образів та методи інтелектуального аналізу.

Наукова новизна отриманих результатів. Запропоновано алгоритм класифікації задач для ЗДР, який є складовою системи автоматичного розв'язання сингулярно збурених задач математичної фізики на ЕОМ, та концепцію реалізації алгоритму методами інтелектуального аналізу.

Апробація результатів дослідження. Результати дослідження доповідалися на VIII Міжнародній науково-технічній конференції «Комп'ютерне моделювання та оптимізація складних систем», 1-3 листопада 2023, Запоріжжя (тема: “Системний аналіз в математичному та інформаційному моделюванні напівпровідникових p-i-n-структур”); XVII Всеукраїнській науково-практичній конференції "Інформаційні технології у професійній діяльності», 5 листопада, 2024, Рівне (тема: “Системи автоматизованого проектування напівпровідникових структур”); VII Міжнародній науково-практичній конференції «Моделювання, керування та інформаційні технології (MCIT–2024)», 7–9 листопада 2024, Рівне (тема: “DEVELOPMENT OF A SOFTWARE ARCHITECTURE CONCEPT FOR SOLVING SINGULARLY PERTURBED PROBLEMS IN SEMICONDUCTOR ELECTRONICS”) [5-7].

Структура роботи. Кваліфікаційна робота складається зі змісту, вступу, трьох розділів, висновків та списку використаних літературних джерел. До роботи додається реферат українською та англійською мовами.

Загальний обсяг роботи складає 54 сторінок. Робота містить 6 рисунків, 1 лістинг з фрагментами коду програми. Список використаних літературних джерел складає 33 найменування.

У першому розділі роботи розглядається предметна область дослідження – математичні моделі складних систем (систем, що мають особливості), інструментальні засоби моделювання характеристик таких систем. Наводиться, також, приклад типової математичної моделі напівпровідникової електроніки, проводиться її аналіз методами примежового шару і вказується на ті обставини, які перешкоджають перетворенню відповідних автоматизованих алгоритмів розв’язання такого типу задач в автоматичні (без участі людини у процесі реалізації алгоритму).

У другому розділі роботи розглядаються найбільш відомі методи класифікації та ідентифікації інформаційних об’єктів. Приводиться загальна постановка задачі класифікації, розглядаються загальні етапи її розв’язання, згадуються методи класифікації, які ґрунтуються на теоретико-множинному

підході, статистичному підході, методах машинного навчання. Також надано інформацію про способи оцінки ефективності результатів класифікації. Увага приділяється розробці інформаційної моделі задачі Коші / крайової задачі для звичайного диференціального рівняння, визначенню ознак для ідентифікації відповідних задач.

Третій розділ роботи присвячено методам глибокого навчання, що спрямовані на вирішення задачі ідентифікації математичних постановок задач. Зроблено огляд загальних теоретичних положень глибокого навчання, огляд найбільш відомих типів нейронних мереж та їх архітектури. Запропоновано концепцію розв'язання проблеми ідентифікації постановок задач (крайових/з початковими умовами) для звичайних диференціальних рівнянь. У перспективі алгоритм ідентифікації типів відповідних математичних задач забезпечить автоматизацію процесу розв'язання складних модельних задач символьно-числовими методами.

РОЗДІЛ 1

СИСТЕМИ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ. ПОТОЧНИЙ СТАН І ПЕРСПЕКТИВИ РОЗВИТКУ

1.1. Інформаційні системи на робочому місці математика

Системи символної (комп'ютерної) математики чи комп'ютерної алгебри - це програмні продукти здатні вирішувати широке коло задач різного рівня обчислювальної складності: надають можливість формулювати задачу за допомогою спеціальної символної мови в термінах математики; автоматизувати виконання як чисельних, так і аналітичних обчислень; розробляти складні обчислювальні алгоритми розв'язання задач; застосовувати засоби візуалізації процесів та даних тощо [8,9]. Відмітимо, що це поняття включає не тільки сукупність сучасних програмних й апаратних засобів, а й теоретичних і технологічних.

Системи комп'ютерної математики можна зарахувати до категорії інженерних програм автоматизованого проектування таких, як CAD (Computer Aided Design - програмні засоби для автоматизованого проектування), CAM (Computer Aided Manufacturing – системи підготовки технологічного процесу виробництва), орієнтована на використання EOM), CAE (Computer Aided Engeneering – загальна назва програм або програмних пакетів, призначених для інженерних розрахунків, аналізу і симуляції фізичних процесів), EDA (Electronic Design Automation – САПР електронних пристроїв, радіоелектронних засобів, інтегральних схем, друкованих плат). Математичні пакети є невід'ємною складовою систем автоматизованого проектування.

До 90-х років XX ст. САПР вважалися суто професійними. В середині 90-х відбувся переломний момент для світового ринку CAD/CAM/CAE/EDA. Ці системи почали використовуватись масово. Відповідні засоби моделювання з промисловими можливостями стали доступні користувачам персональних комп'ютерів. Відбулася інтеграція символних обчислень і чисельних розрахунків.

Комп'ютерна математика - новий напрямок науки і техніки, що виник на стику класичної математики та інформатики. У сучасних математичних пакетах застосовується принцип конструювання моделі, а не традиційне «мистецтво програмування». Тобто користувач лише здійснює постановку задачі, а методи та алгоритми її розв'язання система знаходить сама. Більше того, такі рутинні операції, як розкриття дужок, перетворення виразів, знаходження коренів рівнянь, похідних та невизначених інтегралів пакетні алгоритми реалізують у символічному вигляді, причому практично без втручання користувача.

Основна перевага систем комп'ютерної математики полягає у можливості виконання обчислень в аналітичному вигляді та у можливості проведення арифметичних та багатьох інших обчислень практично з будь-якою бажаною точністю та без обмежень за максимальними (мінімальними) значеннями чисел.

Сучасні математичні пакети можна використовувати і як звичайний калькулятор, і як засоби для спрощення виразів під час вирішення будь-яких задач, і як засіб розв'язання інженерних задач, і як генератор графіки. Стандартними стали також засоби взаємодії з мережею Інтернет (генерація HTML-сторінок виконується тепер у процесі обчислень), використання хмарних засобів. Відмітимо, що сучасні програми цього класу відносяться до найбільш складних програмних продуктів, що вимагають для своєї розробки великих інтелектуальних, трудових та фінансових витрат.

Дамо короткий оглядом програм комп'ютерної алгебри. Практично всі сучасні програми комп'ютерної математики (приклади найбільш відомих з понад 30, які використовуються на даний час [10]: Maple, MathCad, Mathematica, MatLab) мають вбудовані функції символічних обчислень. Базовими типами даних у них є числа та математичні вирази. Системи комп'ютерної алгебри працюють наступним чином:

- математичні об'єкти (алгебраїчні вирази, ряди, рівняння, вектори, матриці та ін.) та вказівки на те, що з ними потрібно робити, задаються користувачем вхідною мовою системи у вигляді символічних виразів;

- інтерпретатор аналізує дані та переводить символні вирази у внутрішнє представлення;

- символний процесор системи виконує необхідні перетворення чи обчислення та подає відповідь у математичній нотації.

Дамо короткий перелік функцій систем комп'ютерної алгебри (більш повний можна знайти, наприклад в [8-10]):

- спрощення виразів до меншого розміру або приведення до стандартного вигляду,
- підстановки символних та чисельних значень у виразі,
- зміна виду виразів: розкриття добутків та степенів, часткова та повна факторизація (розкладання на множники),
- розкладання на прості дроби, запис тригонометричних функцій через експоненти, перетворення логічних виразів,
- знаходження границь функцій та послідовностей,
- операції з рядами (підсумовування, множення, суперпозиція),
- диференціювання в частинних та повних похідних,
- знаходження невизначених та визначених інтегралів (символьне інтегрування), вирішення лінійних та нелінійних рівнянь,
- алгебраїчне (нечислове) розв'язання диференціальних та скінченно-різницевих рівнянь,
- інтегральні перетворення.

Спостерігаються наступні тенденції. Сучасні інженерні задачі стають все більш складними. Для їх вирішення потрібно задіювати комплекс математичних методів та відповідних програмних засобів. Складність алгоритму розв'язання на певних етапах моделювання потребує втручання людини (зауважимо, що у різних інженерних задачах етапи моделювання з відповідними проблемами відрізняються). Виникає проблема переведення алгоритмів моделювання складних систем із розряду автоматизованих в розряд автоматичних. У Інституті кібернетики імені В. М. Глушкова НАН України створено інтелектуальну систему комп'ютерної математики (ICKM) InparSolver для автоматичного

розв'язування задач обчислювальної математики з наближеними даними, які виникають при математичному моделюванні складних процесів та явищ у наукових та інженерних дослідженнях, а саме: розв'язання систем лінійних алгебраїчних рівнянь, розв'язання задачі на пошук власних значень, розв'язання систем нелінійних рівнянь, задачі Коші для систем звичайних диференціальних рівнянь з наближеними даними. При цьому автоматично виконується розпаралелення обчислень та даних при ефективному використанні обчислювальних ресурсів комп'ютера [11]. Можливості відповідних інтелектуальних систем, очевидно, потрібно розширювати.

1.2. Математичне моделювання складних систем. Засоби і шляхи підвищення ефективності моделювання

Об'єктом сучасних наукових досліджень зазвичай є складні системи. Вони характеризуються низкою ознак. Основні із них – складаються із великої кількості різнотипних елементів (систем); мають складну структуру, в якій можна виділити зворотні зв'язки; нелінійна динаміка елементів системи; відкритість (доступність для зовнішніх впливів) [12].

1.2.1. Асимптотичні методи в моделюванні складних систем

Один із важливих методологічних підходів до дослідження складних систем є асимптотичний аналіз – метод опису граничної поведінки функцій [1-4, 13-15]. Асимптотичні методи є групою наближених методів розв'язання задач, які (поряд з чисельними методами, частина яких в певній мірі є похідними від асимптотичних) використовуються тоді, коли через істотні особливості досліджуваних систем немає можливості отримати точні аналітичні розв'язки задач моделювання. Такими особливостями є, наприклад, нелінійності, змінні коефіцієнти, межі складної форми та нелінійні граничні умови на відомих або, у деяких випадках, невідомих межах тощо.

Асимптотичний аналіз включає сукупність методів дискретизації, збурень, усереднення, примежового шару, зрощення, Вішіка-Люстерніка, ВБК тощо [14].

Усі методи об'єднує ідея заміни деяких функцій (операторів), що відповідають реальності, асимптотично еквівалентними (в області визначення функції чи її частині) з метою спрощення (проведення декомпозиції) задачі.

Асимптотичний аналіз використовується в прикладній математиці для побудови чисельних методів розв'язання рівнянь, у математичній статистиці та теорії ймовірностей для визначення граничних властивостей випадкових величин та статистичних оцінок, в інформатиці під час аналізу ефективності алгоритмів тощо.

Асимптотичний аналіз – ключовий інструмент вивчення властивостей диференціальних рівнянь, які є математичними моделями явищ реального світу. Як правило, такі моделі містять деякий безрозмірний параметр, який вважають дуже малим (збуренням) у масштабах характеристик задачі, що розглядається. У залежності від того, як малий параметр входить у математичну модель, виділяють регулярно та сингулярно збурені задачі, які є предметом вивчення теорії збурень [1]. Результатом застосування методів теорії збурень є рекурентна послідовність задач нижчого рівня складності (у порівнянні з вихідною задачею), алгоритми розв'язання яких, як правило, відомі. З точки зору системного аналізу відбувається декомпозиція початкової (вихідної) задачі.

1.2.2. Алгоритм розв'язання сингулярно збурених задач напівпровідникової електроніки методом теорії збурень

Алгоритм виділення малого параметра (нормування).

У якості прикладу розглянемо модельну сингулярно збурену нелінійну задачу напівпровідникової електроніки. Математична постановка задачі (з деяким поясненням фізичного змісту параметрів системи) має наступний вигляд:

$$\begin{cases} \mu^2 \frac{d\tilde{E}}{d\tilde{x}} = (\tilde{p} - \tilde{n} + N_d), \\ \frac{d^2 \tilde{n}}{d\tilde{x}^2} + \frac{d\tilde{n}}{d\tilde{x}} \tilde{E} + \tilde{n} \frac{d\tilde{E}}{d\tilde{x}} - A_n \tilde{n} = 0, \\ \frac{d^2 \tilde{p}}{d\tilde{x}^2} - \frac{d\tilde{p}}{d\tilde{x}} \tilde{E} - \tilde{p} \frac{d\tilde{E}}{d\tilde{x}} - A_p \tilde{p} = 0, \end{cases} \quad (1.1)$$

$$\left. \frac{d\tilde{n}}{d\tilde{x}} - \gamma_n w \tilde{n} \right|_{\tilde{x}=0} = \frac{J}{eD_n} \frac{w}{N_i}, \quad \left. -\frac{d\tilde{p}}{d\tilde{x}} - \gamma_p w \tilde{p} \right|_{\tilde{x}=0} = 0, \quad (1.2)$$

$$\left. \frac{d\tilde{n}}{d\tilde{x}} - \gamma_n w \tilde{n} \right|_{\tilde{x}=1} = 0, \quad \left. -\frac{d\tilde{p}}{d\tilde{x}} - \gamma_p w \tilde{p} \right|_{\tilde{x}=1} = \frac{J}{eD_p} \frac{w}{N_i},$$

$$\tilde{E}|_{\tilde{x}=0} = 0, \quad \tilde{E}|_{\tilde{x}=1} = 0, \quad \int_0^1 \tilde{E} d\tilde{x} = \tilde{U}, \quad (1.3)$$

де $\tilde{E}(\tilde{x})$, $\tilde{n}(\tilde{x})$, $\tilde{p}(\tilde{x})$ – шукані функції напруженості електричного поля, розподілів концентрацій електронів і дірок в активній області р-і-п-діода відповідно (використовується нормування $\tilde{x} = \frac{x}{w}$, $\tilde{E} = \frac{Eew}{kT}$, $\tilde{n} = \frac{n}{N_i}$, $\tilde{p} = \frac{p}{N_i}$);

$\mu^2 = \frac{\varepsilon \varepsilon_0 kT}{e^2 w^2 N_i}$ (малий параметр $\mu \sim 10^{-6} \div 10^{-8}$), ε – відносна діелектрична стала, ε_0 –

діелектрична стала, w – характерний розмір активної області діода, e – заряд електрона, N_i – концентрація носіїв заряду у власному напівпровіднику, k – стала

Больцмана, T – температура; $A_n = \frac{w^2}{D_n \tau_n^*}$, $A_p = \frac{w^2}{D_p \tau_p^*}$, D_n , D_p – коефіцієнти дифузії

дірок та електронів відповідно, τ_n^* , τ_p^* – характерні часи рекомбінації носіїв заряду

в об'ємі активної області (у даній математичній моделі є сталими, які залежать, як і коефіцієнти дифузії, від обраного матеріалу напівпровідника); $N_d = N_d(\tilde{x})$ –

профіль легування; J – густина струму інжекції, $\tilde{U}$ – різниця потенціалів на активній області, γ_n , γ_p – коефіцієнти рекомбінації носіїв заряду на контактах. У

подальших викладках символ “~” опускається.

Математична модель (1.1)-(1.3) описує стаціонарний процес формування електронно-діркової плазми в одновимірній області (активній або і-області) р-і-п-діодів, які являють собою тришарову (у найпростішому випадку) структуру напівпровідникових матеріалів з різними типами провідності. Система (1.1) містить рівняння Пуассона та рівняння неперервності електронного та діркового струмів. Співвідношення (1.2), (1.3) визначають умови на межах просторової області: визначається потік (густина струму) заряджених частинок на межах області та відповідна різниця потенціалів (напруга).

У такій постановці отримуємо сингулярно збурену крайову задачу для нелінійної системи диференціальних рівнянь [3, 4, 16].

Алгоритм декомпозиції.

Систему рівнянь (1.1) подамо у наступному вигляді:

$$\mu^2 \frac{dE}{dx} = F_E(p(x), n(x)), \quad (1.4)$$

$$\frac{d^2 n}{dx^2} - A_n n = F_n(n(x), E(x)),$$

$$\frac{d^2 p}{dx^2} - A_p p = F_p(p(x), E(x)).$$

Аналогічно до [17,18] пропонуємо шукати розв'язок системи рівнянь у вигляді:

$$n(x) = N(x) + \underline{N}(\underline{\xi}) + \overline{N}(\overline{\xi}), \quad (1.5)$$

$$p(x) = P(x) + \underline{P}(\underline{\xi}) + \overline{P}(\overline{\xi}),$$

$$E(x) = \tilde{E}(x) + \underline{E}(\underline{\xi}) + \overline{E}(\overline{\xi}),$$

де $N(x)$, $P(x)$, $\tilde{E}(x)$ - регулярні частини асимптотик; $\underline{N}(\underline{\xi})$, $\underline{P}(\underline{\xi})$, $\underline{E}(\underline{\xi})$, $\overline{N}(\overline{\xi})$, $\overline{P}(\overline{\xi})$, $\overline{E}(\overline{\xi})$ - примежові поправки асимптотик відповідно в околах точок $x=0$

та $x=l$ ($\underline{\xi} = \frac{x}{\mu}$, $\overline{\xi} = \frac{1-x}{\mu}$ - регуляризуючі розтяги).

Асимптотичні ряди пропонуємо будувати у наступному вигляді:

$$\begin{pmatrix} n(x, \mu) \\ p(x, \mu) \\ E(x, \mu) \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^m \mu^i n_i(x) \\ \sum_{i=0}^m \mu^i p_i(x) \\ \sum_{i=0}^m \mu^i E_i(x) \end{pmatrix} + \begin{pmatrix} \sum_{i=0}^m \mu^i \underline{N}_i(\underline{\xi}) \\ \sum_{i=0}^m \mu^i \underline{P}_i(\underline{\xi}) \\ \sum_{i=-1}^m \mu^i \underline{E}_i(\underline{\xi}) \end{pmatrix} + \begin{pmatrix} \sum_{i=0}^m \mu^i \overline{N}_i(\overline{\xi}) \\ \sum_{i=0}^m \mu^i \overline{P}_i(\overline{\xi}) \\ \sum_{i=-1}^m \mu^i \overline{E}_i(\overline{\xi}) \end{pmatrix} + \begin{pmatrix} R_{n(m)}(x, \mu) \\ R_{p(m)}(x, \mu) \\ R_{E(m)}(x, \mu) \end{pmatrix}, \quad (1.6)$$

де $R_{E(m)}(x, \mu)$, $R_{n(m)}(x, \mu)$, $R_{p(m)}(x, \mu)$ - залишкові члени.

Підстановка (1.5) у (1.4) та виділення регулярних складових рівнянь і примежових поправок способом, який аналогічний тому, що викладено у [3,4], дозволяє переписати систему рівнянь (1.4).

Таким чином, рівняння Пуассона набуває вигляду:

$$\begin{aligned} \mu^2 \frac{d(\tilde{E}(x) + \underline{E}(\underline{\xi}) + \overline{E}(\overline{\xi}))}{dx} &= F_E(P(x) + \underline{P}(\underline{\xi}) + \overline{P}(\overline{\xi}), N(x) + \underline{N}(\underline{\xi}) + \overline{N}(\overline{\xi})) = \\ &= \tilde{F}_E(x) + \underline{F}_E(\underline{\xi}) + \overline{F}_E(\overline{\xi}), \end{aligned} \quad (1.7.1)$$

де

$$\begin{aligned} \tilde{F}_E(x) &= F_E(P(x), N(x)) = P(x) - N(x), \\ \underline{F}_E(\underline{\xi}) &= F_E(P(\underline{\mu\xi}) + \underline{P}(\underline{\xi}), N(\underline{\mu\xi}) + \underline{N}(\underline{\xi})) - F_E(P(\underline{\mu\xi}), N(\underline{\mu\xi})) = \underline{P}(\underline{\xi}) - \underline{N}(\underline{\xi}), \\ \overline{F}_E(\overline{\xi}) &= F_E(P(1 - \underline{\mu\xi}) + \overline{P}(\overline{\xi}), N(1 - \underline{\mu\xi}) + \overline{N}(\overline{\xi})) - F_E(P(1 - \underline{\mu\xi}), N(1 - \underline{\mu\xi})) = \\ &= \overline{P}(\overline{\xi}) - \overline{N}(\overline{\xi}). \end{aligned}$$

Відповідно, рівняння неперервності струму електронів має наступну структуру:

$$\begin{aligned} \frac{d^2(N(x) + \underline{N}(\underline{\xi}) + \overline{N}(\overline{\xi}))}{dx^2} - A_n(N(x) + \underline{N}(\underline{\xi}) + \overline{N}(\overline{\xi})) &= \\ = F_n(N(x) + \underline{N}(\underline{\xi}) + \overline{N}(\overline{\xi}), \tilde{E}(x) + \underline{E}(\underline{\xi}) + \overline{E}(\overline{\xi})) &= \tilde{F}_n(x) + \underline{F}_n(\underline{\xi}) + \overline{F}_n(\overline{\xi}), \end{aligned} \quad (1.7.2)$$

де

$$\tilde{F}_n(x) = F_n(N(x), E(x)) = -\frac{d}{dx}(N(x)\tilde{E}(x)),$$

$$\begin{aligned}
F_n(\underline{\xi}) &= F_n(N(\underline{\mu\xi}) + \underline{N}(\underline{\xi}), \tilde{E}(\underline{\mu\xi}) + \underline{E}(\underline{\xi})) - F_n(N(\underline{\mu\xi}), \tilde{E}(\underline{\mu\xi})) = \\
&= -\frac{1}{\mu} \frac{d}{d\underline{\xi}} \left((N(\underline{\mu\xi}) + \underline{N}(\underline{\xi})) (\tilde{E}(\underline{\mu\xi}) + \underline{E}(\underline{\xi})) \right) + \frac{1}{\mu} \frac{d}{d\underline{\xi}} (N(\underline{\mu\xi}) \tilde{E}(\underline{\mu\xi})) = \\
&= -\frac{1}{\mu} \frac{d}{d\underline{\xi}} (\underline{N}(\underline{\xi}) \tilde{E}(\underline{\mu\xi}) + N(\underline{\mu\xi}) \underline{E}(\underline{\xi}) + \underline{N}(\underline{\xi}) \underline{E}(\underline{\xi})), \\
\overline{F}_n(\overline{\xi}) &= F_n(N(1 - \underline{\mu\xi}) + \overline{N}(\overline{\xi}), \tilde{E}(1 - \underline{\mu\xi}) + \overline{E}(\overline{\xi})) - F_n(N(1 - \underline{\mu\xi}), \tilde{E}(1 - \underline{\mu\xi})) = \\
&= \frac{1}{\mu} \frac{d}{d\overline{\xi}} \left((N(1 - \underline{\mu\xi}) + \overline{N}(\overline{\xi})) (\tilde{E}(1 - \underline{\mu\xi}) + \overline{E}(\overline{\xi})) \right) - \frac{1}{\mu} \frac{d}{d\overline{\xi}} (N(1 - \underline{\mu\xi}) \tilde{E}(1 - \underline{\mu\xi})) = \\
&= \frac{1}{\mu} \frac{d}{d\overline{\xi}} (\overline{N}(\overline{\xi}) \tilde{E}(1 - \underline{\mu\xi}) + N(1 - \underline{\mu\xi}) \overline{E}(\overline{\xi}) + \overline{N}(\overline{\xi}) \overline{E}(\overline{\xi})).
\end{aligned}$$

Аналогічно до (1.7.2) запишемо рівняння неперервності струму дірок:

$$\begin{aligned}
&\frac{d^2(P(x) + \underline{P}(\underline{\xi}) + \overline{P}(\overline{\xi}))}{dx^2} - A_p(P(x) + \underline{P}(\underline{\xi}) + \overline{P}(\overline{\xi})) = \\
&= F_p(P(x) + \underline{P}(\underline{\xi}) + \overline{P}(\overline{\xi}), \tilde{E}(x) + \underline{E}(\underline{\xi}) + \overline{E}(\overline{\xi})) - \tilde{F}_p(x) + \underline{F}_p(\underline{\xi}) + \overline{F}_p(\overline{\xi}), (1.7.3)
\end{aligned}$$

де

$$\begin{aligned}
\tilde{F}_p(x) &= F_p(P(x), E(x)) = \frac{d}{dx} (P(x) \tilde{E}(x)), \\
\underline{F}_p(\underline{\xi}) &= F_p(P(\underline{\mu\xi}) + \underline{P}(\underline{\xi}), \tilde{E}(\underline{\mu\xi}) + \underline{E}(\underline{\xi})) - F_p(P(\underline{\mu\xi}), \tilde{E}(\underline{\mu\xi})) = \\
&= \frac{1}{\mu} \frac{d}{d\underline{\xi}} \left((P(\underline{\mu\xi}) + \underline{P}(\underline{\xi})) (\tilde{E}(\underline{\mu\xi}) + \underline{E}(\underline{\xi})) \right) - \frac{1}{\mu} \frac{d}{d\underline{\xi}} (P(\underline{\mu\xi}) \tilde{E}(\underline{\mu\xi})) = \\
&= \frac{1}{\mu} \frac{d}{d\underline{\xi}} (\underline{P}(\underline{\xi}) \tilde{E}(\underline{\mu\xi}) + P(\underline{\mu\xi}) \underline{E}(\underline{\xi}) + \underline{P}(\underline{\xi}) \underline{E}(\underline{\xi})), \\
\overline{F}_p(\overline{\xi}) &= F_p(P(1 - \underline{\mu\xi}) + \overline{P}(\overline{\xi}), \tilde{E}(1 - \underline{\mu\xi}) + \overline{E}(\overline{\xi})) - F_p(P(1 - \underline{\mu\xi}), \tilde{E}(1 - \underline{\mu\xi})) = \\
&= -\frac{1}{\mu} \frac{d}{d\overline{\xi}} \left((P(1 - \underline{\mu\xi}) + \overline{P}(\overline{\xi})) (\tilde{E}(1 - \underline{\mu\xi}) + \overline{E}(\overline{\xi})) \right) + \frac{1}{\mu} \frac{d}{d\overline{\xi}} (P(1 - \underline{\mu\xi}) \tilde{E}(1 - \underline{\mu\xi})) = \\
&= -\frac{1}{\mu} \frac{d}{d\overline{\xi}} (\overline{P}(\overline{\xi}) \tilde{E}(1 - \underline{\mu\xi}) + P(1 - \underline{\mu\xi}) \overline{E}(\overline{\xi}) + \overline{P}(\overline{\xi}) \overline{E}(\overline{\xi})).
\end{aligned}$$

Виконаємо підстановку (1.6) у (1.7) з наступним розкладом отриманих рівнянь за степенями малого параметра. Аналогічний алгоритм перетворень застосовуємо до граничних умов (1.2), (1.3). Отримаємо наступну рекурентну послідовність задач:

$$\begin{cases} p_0(x) - n_0(x) = 0, \\ \frac{d^2 n_0(x)}{dx^2} - A_n n_0(x) = -\frac{d}{dx}(n_0(x) E_0(x)), \\ \frac{d^2 p_0(x)}{dx^2} - A_p p_0(x) = \frac{d}{dx}(p_0(x) E_0(x)). \end{cases} \quad (1.8.1)$$

$$\begin{aligned} \left. \frac{dn_0(x)}{dx} - \gamma_n w n_0(x) \right|_{x=0} &= \frac{J}{eD_n} \frac{w}{N_i}, \\ -\left. \frac{dp_0(x)}{dx} - \gamma_p w p_0(x) \right|_{x=1} &= \frac{J}{eD_p} \frac{w}{N_i}. \end{aligned}$$

$$\begin{cases} \frac{d \underline{E}_{-1}(\underline{\xi})}{d \underline{\xi}} = \underline{P}_0(\underline{\xi}) - \underline{N}_0(\underline{\xi}), \\ \frac{d^2 \underline{N}_0(\underline{\xi})}{d \underline{\xi}^2} = -\frac{d}{d \underline{\xi}} \left((n_0(0) + \underline{N}_0(\underline{\xi})) E_{-1}(\underline{\xi}) \right), \\ \frac{d^2 \underline{P}_0(\underline{\xi})}{d \underline{\xi}^2} = \frac{d}{d \underline{\xi}} \left((p_0(0) + \underline{P}_0(\underline{\xi})) E_{-1}(\underline{\xi}) \right). \end{cases} \quad (1.8.2)$$

$$\left. \underline{E}_{-1}(\underline{\xi}) \right|_{x=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \underline{E}_{-1}(\underline{\xi}) = E^*,$$

$$\left. \frac{d \underline{N}_0(\underline{\xi})}{d \underline{\xi}} \right|_{x=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \underline{N}_0(\underline{\xi}) = 0,$$

$$\left. \frac{d \underline{P}_0(\underline{\xi})}{d \underline{\xi}} \right|_{x=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \underline{P}_0(\underline{\xi}) = 0.$$

$$\left\{ \begin{array}{l} -\frac{d\bar{E}_{-1}(\bar{\xi})}{d\bar{\xi}} = \bar{P}_0(\bar{\xi}) - \bar{N}_0(\bar{\xi}), \\ \frac{d^2\bar{N}_0(\bar{\xi})}{d\bar{\xi}^2} = \frac{d}{d\bar{\xi}} \left(\left(n_0(1) + \bar{N}_0(\bar{\xi}) \right) \bar{E}_{-1}(\bar{\xi}) \right), \\ \frac{d^2\bar{P}_0(\bar{\xi})}{d\bar{\xi}^2} = -\frac{d}{d\bar{\xi}} \left(\left(p_0(1) + \bar{P}_0(\bar{\xi}) \right) \bar{E}_{-1}(\bar{\xi}) \right). \end{array} \right. \quad (1.8.3)$$

$$\bar{E}_{-1}(\bar{\xi}) \Big|_{x=1} = 0, \quad \lim_{\bar{\xi} \rightarrow \infty} \bar{E}_{-1}(\bar{\xi}) = E^*,$$

$$\frac{d\bar{N}_0(\bar{\xi})}{d\bar{\xi}} \Big|_{x=1} = 0, \quad \lim_{\bar{\xi} \rightarrow \infty} \bar{N}_0(\bar{\xi}) = 0, \quad \frac{d\bar{P}_0(\bar{\xi})}{d\bar{\xi}} \Big|_{x=1} = 0, \quad \lim_{\bar{\xi} \rightarrow \infty} \bar{P}_0(\bar{\xi}) = 0.$$

$$\left\{ \begin{array}{l} p_1(x) - n_1(x) = 0, \\ \frac{d^2 n_1(x)}{dx^2} - A_n n_1(x) = -\frac{d}{dx} \left(n_1(x) E_0(x) + n_0(x) E_1(x) \right), \\ \frac{d^2 p_1(x)}{dx^2} - A_p p_1(x) = \frac{d}{dx} \left(p_1(x) E_0(x) + p_0(x) E_1(x) \right). \end{array} \right. \quad (1.9.1)$$

$$\frac{dn_1(x)}{dx} - \gamma_n w n_1(x) \Big|_{x=0} = 0,$$

$$-\frac{dp_1(x)}{dx} - \gamma_p w p_1(x) \Big|_{x=1} = 0.$$

$$\left\{ \begin{array}{l} \frac{d\underline{E}_0(\underline{\xi})}{d\underline{\xi}} = \underline{P}_1(\underline{\xi}) - \underline{N}_1(\underline{\xi}), \\ \frac{d^2 \underline{N}_1(\underline{\xi})}{d\underline{\xi}^2} = -\frac{d}{d\underline{\xi}} \left(E_0(0) \underline{N}_0(\underline{\xi}) + \underline{E}_0(\underline{\xi}) \left(n_0(0) + \underline{N}_0(\underline{\xi}) \right) + \right. \\ \quad \left. + \underline{E}_{-1}(\underline{\xi}) \left(n_1(0) + \underline{\xi} n'_0(0) \right) + \underline{N}_1(\underline{\xi}) \underline{E}_{-1}(\underline{\xi}) \right), \\ \frac{d^2 \underline{P}_1(\underline{\xi})}{d\underline{\xi}^2} = \frac{d}{d\underline{\xi}} \left(E_0(0) \underline{P}_0(\underline{\xi}) + \underline{E}_0(\underline{\xi}) \left(p_0(0) + \underline{P}_0(\underline{\xi}) \right) + \right. \\ \quad \left. + \underline{E}_{-1}(\underline{\xi}) \left(p_1(0) + \underline{\xi} p'_0(0) \right) + \underline{P}_1(\underline{\xi}) \underline{E}_{-1}(\underline{\xi}) \right). \end{array} \right. \quad (1.9.2)$$

$$\begin{aligned}
& \left(E_0(\mu \underline{\xi}) + \underline{E}_0(\underline{\xi}) \right) \Big|_{\underline{\xi}=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \underline{E}_0(\underline{\xi}) = 0, \\
& \left. \frac{d \underline{N}_1(\underline{\xi})}{d \underline{\xi}} - \gamma_n w \underline{N}_0(\underline{\xi}) \right|_{\underline{\xi}=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \underline{N}_1(\underline{\xi}) = 0, \\
& \left. -\frac{d \underline{P}_1(\underline{\xi})}{d \underline{\xi}} - \gamma_p w \underline{P}_0(\underline{\xi}) \right|_{\underline{\xi}=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \underline{P}_1(\underline{\xi}) = 0. \\
& \left\{ \begin{aligned} & -\frac{\partial \bar{E}_0(\bar{\xi})}{\partial \bar{\xi}} = \bar{P}_1(\bar{\xi}) - \bar{N}_1(\bar{\xi}), \\ & \frac{\partial^2 \bar{N}_1(\bar{\xi})}{\partial \bar{\xi}^2} = \frac{d}{d \bar{\xi}} \left(E_0(1) \bar{N}_0(\bar{\xi}) + \bar{E}_0(\bar{\xi}) (n_0(1) + \bar{N}_0(\bar{\xi})) + \right. \\ & \quad \left. + \bar{E}_{-1}(\bar{\xi}) (n_1(1) - \bar{\xi} n'_q(1)) + \bar{N}_1(\bar{\xi}) \bar{E}_{-1}(\bar{\xi}) \right), \\ & \frac{\partial^2 \bar{P}_1(\bar{\xi})}{\partial \bar{\xi}^2} = -\frac{d}{d \bar{\xi}} \left(E_0(1) \bar{P}_0(\bar{\xi}) + \bar{E}_0(\bar{\xi}) (p_0(1) + \bar{P}_0(\bar{\xi})) + \right. \\ & \quad \left. + \bar{E}_{-1}(\bar{\xi}) (p_1(1) - \bar{\xi} p'_q(1)) + \bar{P}_1(\bar{\xi}) \bar{E}_{-1}(\bar{\xi}) \right). \end{aligned} \right. \quad (1.9.3) \\
& \left(E_0(1 - \mu \underline{\xi}) + \bar{E}_0(\underline{\xi}) \right) \Big|_{\underline{\xi}=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \bar{E}_0(\underline{\xi}) = 0, \\
& \left. \frac{d \bar{N}_1(\underline{\xi})}{d \underline{\xi}} - \gamma_n w \bar{N}_0(\underline{\xi}) \right|_{\underline{\xi}=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \bar{N}_1(\underline{\xi}) = 0, \\
& \left. -\frac{d \bar{P}_1(\underline{\xi})}{d \underline{\xi}} - \gamma_p w \bar{P}_0(\underline{\xi}) \right|_{\underline{\xi}=0} = 0, \quad \lim_{\underline{\xi} \rightarrow \infty} \bar{P}_1(\underline{\xi}) = 0.
\end{aligned}$$

Для пошуку членів асимптотик з індексами >1 формуються постановки задач, які аналогічні (1.8), (1.9).

Етап розв'язання послідовності задач.

Явний вигляд функціональних залежностей, які формують асимптотичні ряди за степенями малого параметра задачі (1.1)-(1.3), знаходимо в результаті розв'язання відповідних крайових задач для систем диференціальних рівнянь. Деякі із задач можна розв'язати аналітичними методами, але в основному

приходиться мати справу із такими, частинні розв'язки яких можна знайти універсальними чисельними методами [19]. Відмітимо, що на відміну від моделі початкової постановки задачі, отримані в результаті розчеплення модельні задачі є лінійними, але (у загальному випадку) неоднорідними і з змінними коефіцієнтами.

Алгоритм синтезу розв'язків.

Результат розв'язання задачі родається у вигляді асимтотик, побудованих у вигляді степеневих рядів за малим параметром задачі (μ). У тому випадку, коли частина розв'язків задач типу (1.8)-(1.9) подається у вигляді аналітичних функцій, а частина – у табличному вигляді, потрібно приймати рішення про приведення результатів до єдиного виду подання. Алгоритм синтезу результату, таким чином, може містити процедуру апроксимації.

1.2.3. Проблема автоматизації процесу розв'язання задач асимптотичними методами

Розглянутий у п.1.2.2 роботи приклад розв'язання сингулярно збурених задач є типовим для опису фізичних, соціальних, інформаційних тощо процесів і систем, математичні моделі яких подаються у гідродинамічному наближенні. Відповідно є зміст у побудові уніфікованого автоматичного алгоритму вирішення такого типу задач (без втручання у процес розв'язання людини).

Виділимо етапи розв'язання задачі: формування постановки задачі; виконання процедури нормування і виділення малого параметра задачі; визначення типу перетвореної задачі і прийняття рішення про застосування конкретного методу розв'язання; проведення процедури розчеплення задачі (рівнянь початкової системи і відповідних граничних умов); формування підзадач отриманої рекурентної послідовності; визначення типів підзадач рекурентної послідовності і прийняття рішень про вибір аналітичного чи чисельного методу розв'язання (відмітимо, що перевага тут надається методам, що орієнтовані на символічні перетворення для зменшення впливу обчислювальної похибки на результат), синтез (агрегація) результуючих

асимптотик. Очевидно, що частина із зазначених етапів розв'язання складних модельних задач не може бути реалізованою без втручання людини на даному етапі розвитку відповідних програмних систем.

Напрошується висновок про залучення методів інтелектуального аналізу [20-22] на певних етапах моделювання складних систем, які потребують прийняття рішень, з метою підвищення ефективності роботи відповідних алгоритмів.

На завершення відмітимо переваги методів асимптотичного аналізу (у порівнянні, наприклад, із чисельними методами). Насамперед потрібно зазначити те, що асимптотичні методи приводять до розчеплення початкової задачі і, цим понижують рівень її складності. Це, в свою чергу, забезпечує результативність отриманих алгоритмів. Розчеплені задачі у наглядній формі демонструють особливості протікання процесів у складній системі, «сили» взаємовпливів процесів між собою тощо. Останнє особливо важливе у випадках, коли у математичній моделі потрібно врахувати низку факторів впливу на досліджуваний процес (наприклад, температури, зовнішніх фізичних полів (температурного, електромагнітного), тощо). Через можливість використання символічних перетворень в алгоритмі розв'язання складних задач асимптотичними методами (методами теорії збурень) загальна похибка розв'язку може бути невеликою.

РОЗДІЛ 2

КЛАСИФІКАЦІЯ ТА ІДЕНТИФІКАЦІЯ ЗАДАЧ ДЛЯ ЗВИЧАЙНИХ ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ

Для реалізації задачі автоматичного вибору методу розв’язання початкової/крайової задачі для диференціальних рівнянь (вибіру єдиного методу із відповідної множини методів) потрібно вирішити задачу ідентифікації типу задачі.

2.1. Подання вхідних даних. Опис системи ознак

Для вирішення проблеми автоматичного розв’язання сингулярно збурених задач за схемою п.1.2.2 потрібно розробити алгоритм визначення типів модельних задач для диференціальних рівнянь у рекурентній послідовності (наприклад, (1.8)-(1.9)), яку отримано у результаті відповідних символьних перетворень. Вхідними даними для такого алгоритму є рядки, що містять запис у символьному вигляді звичайних диференціальних рівнянь (системи рівнянь) та початкових (граничних умов).

Рядки вхідних даних мають певні склад і структуру [23] – це операнди виду $Q(x) \cdot \left(\frac{d^i y(x)}{dx^i} \right)^j$ (де $y(x)$ – шукана функція, i – порядок похідної, j – степінь (в лінійних задачах $j=0,1$)), які розділені знаками операцій ‘-’, ‘+’, ‘=’. За допомогою алгоритму відповідних операцій символьних перетворень склад рядка (рівняння/умови) встановлюється автоматично.



Рисунок 2.1. Призначення інформаційної моделі

Вважаємо, що після виділення структурних елементів для подальшого аналізу вхідної інформації доцільним є формування інформаційної моделі (ІМ)

вхідних даних. Інформаційна модель є цифровим двійником (образом) відповідної математичної моделі і відображає найбільш суттєві властивості об'єкта вивчення. ІМ дає відповідь на питання: як повинна подаватись інформація для обробки у системах інтелектуального аналізу?

Пропонуємо інформаційну модель в нашому випадку подавати у вигляді масиву, елементами якого є набір ознак математичного виразу (набір характеристик). Початкові/граничні умови задачі також входять до складу інформаційної моделі. Відмітимо: запропонована структура інформаційної моделі повинна бути динамічною для забезпечення перспективи її розвитку.

Питання вибору ознак є досить важливим. Спрощує дану процедуру та обставина, що ознаки задач для звичайних диференціальних рівнянь добре опрацьовані у теорії звичайних диференціальних рівнянь [24]. Ключовою ознакою диференціального рівняння є наявність у структурі даних похідної

шуканої функції певного порядку $\left(\frac{d^i y(\dots)}{dx^i} \right)^j$. Проте ця ознака є необхідною, але

не достатньою. Для того, щоб рівняння було звичайним диференціальним до його структури не повинні входити інші операторні перетворення, шукана у рівнянні функція повинна залежати лише від одного аргумента. Наприклад, наявність у структурі виразу інтегральних перетворень, згортки є причиною віднесення рівняння (виразу) до іншого типу.

Приймемо за основу систему ознак звичайних диференціальних ознак, які визначені у теорії таких рівнянь [24]. Отже, як сказано вище, у структуру диференціального рівняння повинна входити шукана функція, яка залежить лише від одного аргументу, та її похідні. Диференціальні рівняння поділяються на лінійні та нелінійні; першого, другого, n-го порядку; однорідні та неоднорідні; з сталими та змінними коефіцієнтами тощо [25].

Аналогічним чином розглядаються додаткові умови які забезпечують єдиність розв'язку поставленої задачі (згідно теореми Коші [24]). Додаткові умови визначають значення шуканої функції у певній (визначеній) точці відрізка

інтегрування, а також похідні цієї функції до $(n-1)$ порядку, де n – порядок рівняння. Додаткові умови визначаються у різний спосіб. Як правило вони записуються для початкової точки відрізка інтегрування, або на кінцях відрізка інтегрування.

Розглянемо у якості прикладу інформаційну модель задачі Коші для диференціального рівняння виду:

$$Q_2(x) \cdot \frac{d^2 y(x)}{dx^2} + Q_1(x) \cdot \frac{dy(x)}{dx} + Q_0(x) \cdot y(x) = F(x), \quad (2.1)$$

$$y(x_0) = y_0, \quad \left. \frac{dy(x)}{dx} \right|_{x=x_0} = y^0.$$

Відповідна інформаційна модель є СТРУКТУРОЮ, елементи якої зберігають ‘0’ або ‘1’ у залежності від того, відсутня чи наявна певна ознака у математичній моделі (об’єкті).

Визначимо систему ознак:

- не диференціальне рівняння/диференціальне рівняння;
- лінійне/нелінійне;
- однорідне/неоднорідне;
- першого порядку/другого порядку;
- з сталими коефіцієнтами/з змінними коефіцієнтами;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ПЕРШОГО КОЕФІЦІЄНТА;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ДРУГОГО КОЕФІЦІЄНТА;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ТРЕТЬОГО КОЕФІЦІЄНТА;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ПРАВОЇ ЧАСТИНИ РІВНЯННЯ;
- задано початкові умови/граничні умови;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ГРАНИЧНИХ УМОВ.

Таким чином, інформаційна модель крайової задачі (2.1) має наступний вигляд:

$\{1;0;1;1;1;\{xx\};\{xx\};\{xx\};\{xx\};0;\{0;0;0\}\},$

де $\{xx\}$ -структури, які потребують подальшої деталізації (не виключено, що вони містять надлишкову інформацію).

Альтернативним способом представлення інформаційної моделі полягає у наступному. Кожна ознака, яка характеризує задачу Коші (орієнтуємось на запропонований приклад) отримує свій унікальний ідентифікатор (номер), який при виявленні тієї чи іншої ознаки записується у масив даних. Таким чином, інформаційна модель у даному випадку є структурою з варіативним збереженням ознак на відміну від того, як інформація подається у попередньому варіанті де дані чітко структуровані.

Ідентифікатор ознак, наприклад, має наступну структуру:

- не диференціальне рівняння - 00;
- диференціальне рівняння - 01;
- лінійне - 02;
- нелінійне - 03;
- однорідне - 04;
- неоднорідне - 05;
- першого порядку - 06;
- другого порядку - 07;
- з сталими коефіцієнтами - 08;
- з змінними коефіцієнтами - 09;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ПЕРШОГО
КОЕФІЦІЄНТА;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ДРУГОГО
КОЕФІЦІЄНТА;
- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ТРЕТЬОГО
КОЕФІЦІЄНТА;

- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ПРАВОЇ ЧАСТИНИ РІВНЯННЯ;

- задано початкові умови - 10;

- граничні умови - 11;

- СТРУКТУРА ДЛЯ ОПИСУ ОЗНАК ГРАНИЧНИХ УМОВ.

Очевидно, опис класифікатора можна продовжити. Таким чином, інформаційна модель крайової задачі (2.1) у даній системі має наступний вигляд:

{01;02;05;07;09;{xx};{xx};{xx};{xx};10;{xx}}.

Очевидно, що інформаційна модель досліджуваного об'єкта є спрощеним у певному розумінні його представленням, яке може подаватися різними способами. Фактично це є той набір уніфікованих даних, які будуть надаватися автоматизованій системі прийняття рішень для обробки.

2.2. Постановка задач класифікації та ідентифікації

Одним із ключових елементів сучасних автоматизованих систем прийняття рішень є система розпізнавання ситуацій (образів) [26, 27]. Постановка задач класифікації та ідентифікації є основою багатьох алгоритмів машинного навчання та штучного інтелекту.

Декілька слів про те, яке місце займають задачі класифікації та ідентифікації в системі знань напрямку «Комп'ютерні науки». В кібернетиці, як науці про загальну теорію управління, на даний час виокремились такі складові: теорія інформації, теорія систем, теорія розпізнавання образів, теорія прийняття рішень [28]. Теорія розпізнавання образів розробкою відповідних методів закладає основу для вирішення проблеми «комп'ютерного зору» (проблема автоматизованого сприйняття інформації з навколишнього світу).

Розпізнати – означає віднесення об'єкта розпізнавання за певними ознаками до певної (наперед відомої) множини (групи, класу) об'єктів, що мають спільні ознаки. Формування відповідних класів (множин) об'єктів забезпечують методи класифікації (у певних джерелах поняття класифікації і розпізнавання практично ідентичні, бо на етапі формування класів використовуються ті методи,

які ідентичні розпізнаванню). Класи часто утворюють ієрархічну структуру. Ідентифікація – це задача виявлення конкретного екземпляра у класі.

Прикладом задачі класифікації може служити задача виявлення звичайного диференціального рівняння серед математичних конструкцій різного виду. Ідентифікація у зв'язку із сказаним – визначення конкретного типу диференціального рівняння.

Основні етапи та алгоритми, що застосовуються в задачах класифікації та ідентифікації:

1. *Збір даних*: вхідні дані можуть бути у вигляді тексту, зображень, звуків, виразів і т.д.
2. *Передобробка даних*: включає нормалізацію, стандартизацію, видалення шумів і зменшення розмірності.
3. *Вибір ознак*: вибір важливих ознак або створення нових ознак (фічей), які краще описують об'єкти.
4. *Розподіл на тренувальну і тестову вибірку*: дані діляться на тренувальну вибірку для навчання моделі та тестову вибірку для перевірки її якості.
5. *Навчання моделі*: використання алгоритмів машинного навчання для побудови моделі. Найпоширеніші алгоритми:
 - логістична регресія;
 - дерево рішень;
 - опорних векторів (SVM);
 - використання нейронних мереж.
6. *Оцінка якості моделі*: використання метрик, таких як точність (accuracy), повнота (recall), F-міра (F1-score) для оцінки якості моделі.
7. *Крос-валідація*: метод для оцінки якості моделі на різних підмножинах даних.

Кожен крок процесу ідентифікації нагадує етапи класифікації, але деякі з них мають особливості. Наприклад, серед ознак вибирають ту, яка унікальною в даного об'єкта, використовують метод порівняння з еталонним об'єктом тощо.

Для вирішення задачі автоматизації процесу встановлення типу диференціального рівняння з відповідними початковими/граничними умовами класифікації вхідних об'єктів проводити не потрібно – систематизація і класифікація відповідних задач проведена в теорії звичайних диференціальних рівнянь [25]. Ця робота проводилась за аналогією з одним із сучасних методів машинного навчання – методом дерева рішень.

У подальшому зосередимось на проблемі автоматизації процесу ідентифікації.

Формалізована постановка задачі ідентифікації полягає у наступному.

Вхідними даними задачі є:

- X : набір об'єктів або спостережень, які необхідно ідентифікувати;
- Y : набір можливих міток ідентифікації, де кожна мітка представляє унікальний об'єкт або категорію.

Навчальна вибірка:

- Пара (x_i, y_i) , де x_i – вектор ознак об'єкта, а y_i – мітка, що ідентифікує об'єкт x_i .
- $D = \{(x_i, y_i) | i=1, \dots, N\}$: набір пар (x_i, y_i) , де N – кількість навчальних прикладів.

Модель:

- Функція $f: X \rightarrow Y$, яка на основі вектору ознак X передбачає відповідну мітку Y . Для задачі ідентифікації функція $f(x)$ повинна правильно ідентифікувати конкретний об'єкт серед багатьох можливих.

Функція втрат:

- Функція втрат $L(f(x), y)$ вимірює якість передбачення моделі.

Процес навчання:

- Оптимізація параметрів моделі f з використанням методів, таких як стохастичний градієнтний спуск (SGD) або інші алгоритми оптимізації.

- Мета - мінімізувати функцію втрат на навчальній вибірці.

Математично, задача ідентифікації зводиться до мінімізації функції втрат на навчальній вибірці: $\min \sum_{i=1}^N L(f(x_i), y_i)$.

Задача ідентифікації передбачає навчання моделі на відомих даних і її використання для правильного ідентифікування нових, раніше невідомих об'єктів. Це включає процес підбору і оптимізації моделі, щоб забезпечити високу точність та надійність ідентифікації.

2.3. Методи і моделі машинного навчання

Машинне навчання (Machine Learning, ML) [20] – це галузь штучного інтелекту (ШІ), яка зосереджена на розробці алгоритмів і моделей, здатних навчатися на основі даних та робити прогнози або класифікацію без явного програмування. Це один з ключових інструментів для аналізу великих обсягів даних та автоматизації рішень.

На відміну від традиційних методів, де обчислювальні системи запрограмовані на виконання певних задач, ML-моделі мають здатність покращувати свою продуктивність з часом, аналізуючи дані та виявляючи в них закономірності. Це дозволяє ML-системам вирішувати задачі, які складно чи неможливо запрограмувати вручну.

Машинне навчання (МН) як науковий напрямок виросло з досліджень у галузі штучного інтелекту (ШІ, англ. artificial intelligence, AI). У часи становлення ШІ як академічної дисципліни деякі вчені прагнули навчити машини аналізувати дані. Вони застосовували різні символічні методи, а також перші версії того, що пізніше отримало назву “нейронні мережі” – зокрема, перцептрони. Також використовували ймовірнісні методи, особливо в автоматизованих системах медичної діагностики.

Однак зосередженість на логічних підходах, заснованих на знаннях, призвела до розриву між ШІ та машинним навчанням. Імовірнісні системи зазнали проблем зі збором і поданням даних, що мали як теоретичні, так і

практичні аспекти. На початку 1980-х експертні системи стали домінувати в ШІ, а статистичні методи відійшли на другий план. Робота над навчанням на основі знань тривала в межах ШІ, зокрема в індуктивному логічному програмуванні, проте статистично орієнтовані дослідження перемістилися в галузі розпізнавання образів і інформаційного пошуку.

Машинне навчання, реорганізоване та визнане як окрема галузь, почало активно розвиватися в 1990-х роках. Цей напрямок змінив свою мету з досягнення штучного інтелекту на вирішення прикладних задач. Воно перейшло від символічних методів, успадкованих від ШІ, до методів і моделей, запозичених зі статистики, нечіткої логіки та теорії ймовірностей.

Додатково можна відзначити, що розвиток машинного навчання був значною мірою стимульований ростом обчислювальних потужностей і доступністю великих масивів даних, що дозволило використовувати більш складні моделі і алгоритми для вирішення широкого спектру задач у різних галузях науки та техніки. Ці досягнення відкрили нові можливості для застосування машинного навчання в медицині, фінансах, маркетингу, автономних транспортних засобах та багатьох інших сферах.

Цей науковий напрямок продовжує активно розвиватися, інтегруючи новітні технологічні досягнення та знаходячи все нові й нові застосування в реальному світі.

Зупинимось коротко на головних методах машинного навчання.

Кероване навчання (навчання з вчителем (Supervised learning)) є найбільш поширеним типом машинного навчання. Воно передбачає навчання алгоритму на попередньо підготовленому наборі даних, де кожен приклад супроводжується міткою, яка відображає бажаний результат для цього прикладу. На основі таких мічених даних алгоритм навчається встановлювати зв'язок між вхідними даними та відповідними мітками, що дозволяє йому робити прогнози або класифікації для нових, невідомих даних [20].

Існує багато типів алгоритмів керованого навчання, кожен з яких має свої особливості та області застосування. Ось деякі з них.

Лінійна регресія. Призначення: прогнозування числового значення на основі одного або кількох вхідних параметрів. Приклад використання: прогнозування вартості будинку на основі площі, кількості кімнат, розташування тощо.

Логістична регресія. Призначення: класифікація даних на категорії. Приклад використання: визначення ймовірності настання події, наприклад, чи є електронний лист спамом.

Дерева рішень. Призначення: ухвалення рішень на основі набору правил, представлених у вигляді дерева. Приклад використання: класифікація пацієнтів на групи ризику захворювання на основі їхніх медичних показників.

Метод k-найближчих сусідів (KNN). Призначення: класифікація даних на основі схожості з іншими даними в наборі. Приклад використання: рекомендаційні системи, що пропонують фільми на основі вподобань користувачів.

Підтримка векторних машин (SVM). Призначення: класифікація даних та регресія за допомогою знаходження гіперплощини, що розділяє різні класи. Приклад використання: розпізнавання рукописного тексту, де кожна буква або цифра представляє окремий клас.

Перевагами керованого навчання є:

- **висока точність:** завдяки великій кількості мічених даних, алгоритми керованого навчання можуть досягати високої точності;
- **зрозумілість результатів:** легкість інтерпретації результатів, особливо у випадку простих моделей, таких як лінійна регресія або дерева рішень;
- **універсальність:** широкий спектр застосувань у різних галузях, від медицини до фінансів і маркетингу.

Некероване навчання, також відоме як навчання без вчителя (Unsupervised Learning), – це тип машинного навчання, при якому алгоритм працює на наборі даних, що не має міток. Завдання алгоритму полягає у самостійному знаходженні закономірностей та виявленні структур у даних, групуванні об'єктів за

подібними ознаками та, можливо, прогнозуванні подальших подій на основі виявлених патернів.

Відмітимо основні методи некерованого навчання.

Методи кластеризації. До них відносять метод k-середніх (k-means): (алгоритм кластеризації, який розбиває дані на заздалегідь визначену кількість кластерів, де кожен кластер містить об'єкти, схожі між собою); алгоритм наблжених k-середніх (покращена версія методу k-середніх, яка здатна ефективніше обробляти великі набори даних); метод ієрархічної кластеризації (метод, що створює ієрархічне дерево кластерів на основі подібностей між об'єктами).

Методи зниження розмірності. Основні – аналіз головних компонент (Principal Component Analysis, PCA) (метод зниження розмірності, який вибирає найінформативніші ознаки у наборах даних, зберігаючи при цьому якнайбільше інформації); t-SNE (t-distributed Stochastic Neighbor Embedding) - метод для візуалізації високорозмірних даних у меншому просторі.

Методи виявлення аномалій. Використовуються для знаходження незвичайних або відхилених від норми даних, які можуть вказувати на проблеми або особливі випадки.

Може, також, зустрічатись *комбінація* методів керованого та некерованого навчання, відома як *напівкероване навчання* (Semi-Supervised Learning). У цьому підході алгоритм навчається на наборі даних, що містить як мічені, так і немічені приклади. Мічені приклади використовуються для навчання алгоритму, як у керованому навчанні, а немічені дані застосовуються для виявлення закономірностей, як у некерованому навчанні.

Переваги методів напівкерованого навчання полягають у наступному:

- *Ефективність:* використання великої кількості немічених даних для покращення моделі при обмеженій кількості мічених даних.

- *Зниження витрат:* зменшення потреби в мічених даних, що можуть бути дорогими або складними для збирання.

• *Гнучкість*: можливість знаходити приховані структури у даних, які можуть залишатися непоміченими у чисто керованому навчанні.

Навчання з підкріпленням (Reinforcement Learning, RL) – це тип машинного навчання, в якому алгоритм навчається через взаємодію з довкіллям. Відмінністю від інших методів машинного навчання є те, що RL алгоритм не потребує набору даних із мітками. Натомість він отримує «підкріплення» у вигляді «винагороди» або «штрафу» за свої дії та їх результати. Завдяки цьому алгоритм навчається обирати дії, які максимізують сукупну винагороду з часом.

Методи навчання з підкріпленням мають наступні переваги:

- *ефективність у складних задачах* (RL моделі можуть бути застосовані до задач, які важко або неможливо вирішити за допомогою інших методів машинного навчання, таких як управління автономними транспортними засобами, ігрові стратегії тощо);
- *адаптивність* (RL алгоритми можуть адаптуватися до нових умов і завдань, оскільки вони постійно оновлюють свою стратегію на основі отриманого підкріплення);
- *здатність до самонавчання* (RL моделі не вимагають великих наборів даних із мітками, оскільки вони навчаються на основі взаємодії з середовищем і отриманого підкріплення).

Глибоке навчання (Deep Learning) – це спеціалізована підгалузь машинного навчання, яка використовує штучні нейронні мережі для розв’язання складних задач. Відмінність глибокого навчання полягає в здатності моделювати високоабстрактні патерни у даних завдяки багатошаровій архітектурі нейронних мереж. Методам та відповідним технологіям глибокого навчання буде присвячений третій розділ даної роботи.

В результаті аналізу методів машинного навчання приходимо до висновку про те, що вирішення завдання систематизації та класифікації постановок задач для звичайних диференціальних рівнянь природно відбувається методом дерева рішень. І є ілюстрацією застосування даного методу.

Метод дерева рішень є потужним і інтуїтивно зрозумілим методом машинного навчання, який часто використовується для задач класифікації. Це деревоподібна структура, де кожен внутрішній вузол представляє тест на ознаку, кожна гілка – результат тесту, а кожен листовий вузол – клас. Компонентами дерева рішень є:

1. кореневий вузол (початкова точка дерева, що містить весь навчальний набір і вибирає ознаку для першого поділу);
2. внутрішні вузли (вузли, що представляють тести на певні ознаки. Вони розгалужуються на підвузли на основі результатів тесту);
3. гілки (результати тестів, що з'єднують вузли і ведуть до інших вузлів або листових вузлів);
4. листові вузли: (кінцеві вузли дерева, що представляють передбачуваний клас або вихід).

Процес побудови дерева рішень включає наступні *етапи*:

1. вибір ознаки для поділу: на кожному кроці вибирається ознака, яка найбільше зменшує невизначеність або ентропію даних. Методи вибору включають використання:
 - інформаційного приросту (Information Gain) (різниця в ентропії (кількісна міра впорядкованості) до і після поділу даних);
 - коефіцієнта Джіні (Gini Index) (міра чистоти або кількості коректних класифікацій у підмножині даних);
2. розбиття вузла (дані діляться на підгрупи на основі значень вибраної ознаки);
3. повторення процесу – кожен підвузол стає новим вузлом і процес повторюється для нього. Цей процес триває до досягнення одного з умов зупинки:
 - всі об'єкти у вузлі належать до одного класу;
 - немає більше ознак для поділу;
 - досягнуто максимальну глибину дерева.

Відмітимо переваги та недоліки методу дерева рішень. До переваг відносять простоту інтерпретації (дерева рішень є інтуїтивно зрозумілими і легко візуалізуються); можливість обробки як числових, так і категорійних даних; мінімальна потреба в попередній обробці даних. Недоліками методу дерева рішень є схильність до перенавчання (дерева рішень можуть легко перенавчитися на навчальних даних); нестійкість до шуму (зміна вхідних даних може суттєво змінити структуру дерева).

Багатокласова класифікація – це здатність моделі класифікувати вхідні дані більш ніж на два класи. Цей метод схожий на бінарну класифікацію, але дозволяє розширити модель більш загальних сценаріїв. Це актуально у сферах, де категорії чи класи можуть бути множинними. Багатокласова класифікація – це завдання машинного навчання, у якій модель навчається розрізняти вхідні дані кілька класів. Це означає, що ми маємо більше двох класів (категорій) для класифікації. У цьому контексті “клас” – це категорія, до якої модель намагається віднести вхідні дані. Багатокласова класифікація має багато сфер застосувань, починаючи від комп’ютерного зору та розпізнавання образів до аналізу текстових даних. Важливо, що багатокласова класифікація є узагальненням завдання бінарної класифікації, де є лише два класи.

Існує кілька методів та підходів для вирішення завдань багатокласової класифікації:

- One-vs-All: цей метод полягає у створенні бінарних класифікаторів кожного класу;

- Softmax-регресія: це узагальнення логістичної регресії на багатокласовий випадок. Він використовує функцію softmax для перетворення виходів моделі на ймовірність приналежності до різних класів;

- Дерева прийняття рішень та випадковий ліс: дерева прийняття рішень поділяють дані на класи за допомогою деревоподібної структури, а випадковий ліс поєднує кілька дерев для поліпшення точності;

- Нейронні мережі: глибокі нейронні мережі, такі як нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), часто використовуються для

багатокласової класифікації зображень і текстових даних. Ці мережі можуть вивчати складні залежності між даними та класами.

При цьому методі модель навчається вирішувати кілька завдань одночасно. Замість створення окремих моделей для кожного завдання, багатозадачні моделі навчаються з урахуванням усіх завдань одночасно.

2.4. Метрики якості в задачах ідентифікації

Метрики якості в задачах ідентифікації відіграють важливу роль у визначенні ефективності та точності алгоритмів машинного навчання. Ці метрики дозволяють оцінити, наскільки добре модель виконує свої функції, і вказати на області, де можливо покращити її продуктивність. Приведемо опис основних метрик якості.

1. Точність 1 (Accuracy).

- Визначення: частка правильних передбачень серед усіх передбачень.
- Формула: $\text{Точність} = (TP + TN) / (TP + TN + FP + FN)$, де TP (True Positives) - кількість правильних позитивних передбачень, TN (True Negatives) - кількість правильних негативних передбачень, FP (False Positives) - кількість хибних позитивних передбачень, FN (False Negatives) - кількість хибних негативних передбачень.
- Застосовується у випадках з приблизно однаковим числом позитивних і негативних прикладів.

2. Сприйнятливість (Sensitivity) або Повнота (Recall).

- Визначення: частка правильних позитивних передбачень серед усіх справжніх позитивних випадків.
- Формула: $\text{Сприйнятливість} = TP / (TP + FN)$.
- Важлива у випадках, коли важливо виявити всі позитивні випадки, навіть якщо деякі негативні будуть класифіковані як позитивні.

3. Точність 2 (Precision).

- Визначення: частка правильних позитивних передбачень серед усіх передбачених позитивних випадків.

- Формула: $\text{Точність} = \text{TP} / (\text{TP} + \text{FP})$.
- Важлива у випадках, коли потрібно мінімізувати кількість хибних позитивних передбачень.

4. *F1-міра (F1 Score)*.

- Визначення: гармонійне середнє між точністю і сприйнятливістю.
- Формула:
$$\text{F1-міра} = 2 \cdot \text{Точність} \cdot \text{Сприйнятливість} / (\text{Точність} + \text{Сприйнятливість})$$
- Використовується тоді, коли важливо знайти баланс між точністю і сприйнятливістю.

5. *Специфічність (Specificity)*.

- Визначення: частка правильних негативних передбачень серед усіх справжніх негативних випадків.
- Формула: $\text{Специфічність} = \text{TN} / (\text{TN} + \text{FP})$.
- Важлива у випадках, коли потрібно мінімізувати кількість хибних негативних передбачень.

6. *ROC-крива (Receiver Operating Characteristic curve) та AUC (Area Under the Curve)*.

- Визначення: ROC-крива відображає взаємозв'язок між чутливістю та специфічністю при різних порогах класифікації. AUC представляє площу під ROC-кривою.
- Важлива для оцінки моделі в задачах класифікації, незалежно від вибраного порогу.

РОЗДІЛ 3

НЕЙРОМЕРЕЖЕВІ МОДЕЛІ В ЗАДАЧАХ КЛАСИФІКАЦІЇ ТА ІДЕНТИФІКАЦІЇ

3.1. Нейронні мережі. Базові поняття

В останні роки зростає інтерес до використання методів глибокого навчання у різних практичних сферах: комп'ютерний зір [29], обробка текстів, обробка великих масивів даних тощо. Ця тенденція більш помітна в сферах, які орієнтовані на забезпечення масових потреб (наприклад, перевірка та переклад текстів), але очікується, що глибоке навчання стане ефективним засобом вирішення науково-технічних задач.

Нейронні мережі є основною технологією в глибокому навчанні і машинному навчанні. Вони імітують структуру і роботу біологічних нейронних мереж, таких як головний мозок людини, для вирішення складних задач, що включають розпізнавання образів, класифікацію та прогнозування.

Основними компонентами нейронних мереж є *нейрони* (Neurons) - елементи нейронної мережі, які отримують, обробляють і передають інформацію. Кожен нейрон приймає вхідні сигнали, зважує їх, застосовує активаційну функцію і передає вихідний сигнал наступним нейронам.

Шари (Layers) нейронної мережі — це фундаментальні будівельні блоки, які формують архітектуру нейронної мережі. Кожен шар складається з нейронів, які обробляють вхідні сигнали і передають їх далі у мережу. Зазвичай нейронні мережі мають три основні типи шарів: *вхідний шар* (Input Layer), *приховані шари* (Hidden Layers), *вихідний шар* (Output Layer). Перший приймає вхідні дані, причому кожен нейрон у цьому шарі відповідає за прийом однієї ознаки з вхідних даних. Приховані шари обробляють і трансформують інформацію. Вони складаються з нейронів, які обчислюють зважену суму своїх вхідних сигналів, застосовують активаційну функцію і передають результуючий сигнал наступним шарам. Вихідний шар формує остаточні результати. Кількість нейронів у вихідному шарі відповідає кількості класів у задачі класифікації.

Зв'язки між окремими нейронами в системі організовуються за допомогою *синапсів* (Electrical Synapse). В нейронних мережах термін “синапс” використовується як аналогія до біологічних синапсів у нервовій системі, але має своє специфічне значення в контексті штучних нейронів. У нейронних мережах синапси представляють вагові зв'язки між нейронами різних шарів. Характеристиками синапсів є вагові коефіцієнти, які оновлюються при “навчанні” нейронної мережі. Кожен нейрон отримує вхідні сигнали від попередніх нейронів, зважені відповідними вагами. Входи нейронів підсумовуються, і до цієї суми може додаватися порогове значення (Bias). Вихідний сигнал окремого нейрона генерується згідно відповідної функції активації (Activation Functions) – математичних функцій (бувають різних типів), які вводять нелінійності в модель. Схематично нейронну мережу і окремий нейрон показано на рис. 3.1, 3.2.

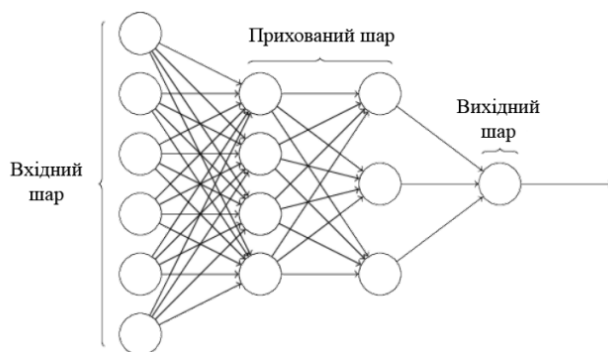


Рисунок 3.1. Схематичне зображення нейронної мережі

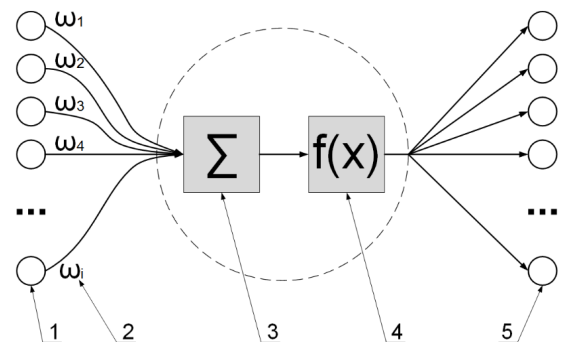


Рисунок 3.2. Модель нейрона (1-входи, 2-ваги, 3-суматор, 4-функція активації, 5-виходи)

Для вирішення конкретної проблеми класифікації нейронна мережа потребує налаштування вагових коефіцієнтів (пройти процедуру навчання). Навчання нейронної мережі – це процес оптимізації ваг і порогів нейронів, щоб мінімізувати помилку прогнозування. Основні етапи включають ініціалізацію, пряме проходження, обчислення функції втрат, зворотне поширення помилки та оновлення ваг. Розглянемо кожен етап детальніше.

Етап ініціалізації. Перед початком навчання ваги нейронів ініціалізуються випадковими значеннями або за допомогою спеціальних методів (наприклад, метод Глорота або Хей). Пороги також ініціалізуються певними значеннями, зазвичай нулями або малими випадковими числами.

Наступний етап навчання отримав назву прямого проходу (Forward Propagation). Відбувається процес перетворення даних у ході проходу інформації через шари мережі. На кожному шарі сигнали множаться на відповідні ваги і підсумовуються. До зваженої суми вхідних сигналів застосовуються активаційні функції (наприклад, ReLU, Sigmoid, Tanh), які перетворюють вхід у вихідний сигнал нейрона. Вихідний шар відображає результати (наприклад, ймовірності віднесення до певного класу у задачі класифікації).

Наступний крок навчання пов'язаний із обчисленням функції втрат (Loss Function). Тобто, визначається помилка між реальними значеннями та прогнозами моделі. Вибір функції втрат залежить від типу задачі:

- крос-ентропія (для задач класифікації);
- середньоквадратична помилка (MSE) (для задач регресії).

Далі обчислюються градієнти функції втрат щодо кожної ваги і порогу (градієнт показує як зміна ваги впливає на помилку). Цей етап отримав назву “зворотне поширення помилки” (Backpropagation).

На завершальному етапі відбувається оновлення ваг (Weight Update). Ваги і пороги оновлюються для мінімізації функції втрат. При цьому використовуються різні алгоритми оптимізації (стохастичний градієнтний спуск (SGD) - оновлення ваг відбувається після кожного прикладу; Adam – адаптивний метод, що використовує моменти градієнтів для більш ефективного навчання; RMSprop – використовує середні квадрати градієнтів для адаптивного масштабування).

Організація навчання відбувається такими способами: виконується один повний прохід всіх навчальних даних через модель (Epochs); дані поділяються на малі групи для обробки (mini-batch gradient descent); дані розподіляються на

навчальну і валідаційну вибірки для оцінки продуктивності моделі під час навчання.

3.2. Моделі нейронних мереж (архітектура НМ)

Моделі нейронних мереж є різноманітними і можуть бути налаштовані для виконання різних задач машинного навчання. Основні типи моделей нейронних мереж включають багатошарові перцептрони, конволюційні нейронні мережі, рекурентні нейронні мережі, генеративно-змагальні мережі. Розглянемо їх архітектуру та основне призначення [21, 30].

Багатошаровий перцептрон (Multilayer Perceptron, MLP, рис.3.1) є однією з найосновніших і найпоширеніших архітектур штучних нейронних мереж. Це класична нейронна мережа, яка складається з декількох шарів нейронів: вхідного шару, одного або більшої кількості прихованих шарів і вихідного шару. Використовується для вирішення широкого спектру задач, включаючи класифікацію та регресію. Архітектура даного типу нейронної мережі має низку переваг: може використовуватися для широкого спектру задач, включаючи класифікацію, регресію, прогнозування; приховані шари автоматично виділяють найважливіші ознаки з вхідних даних; MLP може досягати високої точності, особливо на добре структурованих даних. Недоліки: для ефективного навчання потрібен великий обсяг даних; навчання може займати багато часу, особливо на великих наборах даних і складних моделях; результати сильно залежать від правильного налаштування гіперпараметрів (кількість шарів, кількість нейронів у кожному шарі, швидкість навчання тощо).

Конволюційні (згорткові) нейронні мережі (Convolutional Neural Networks, CNN, рис. 3.3) – це спеціалізовані архітектури нейронних мереж, які широко використовуються для обробки та аналізу зображень і відео. Вони автоматично виділяють ознаки з вхідних даних, що робить їх особливо ефективними для задач розпізнавання образів і класифікації. Особливі елементи такої мережі, –

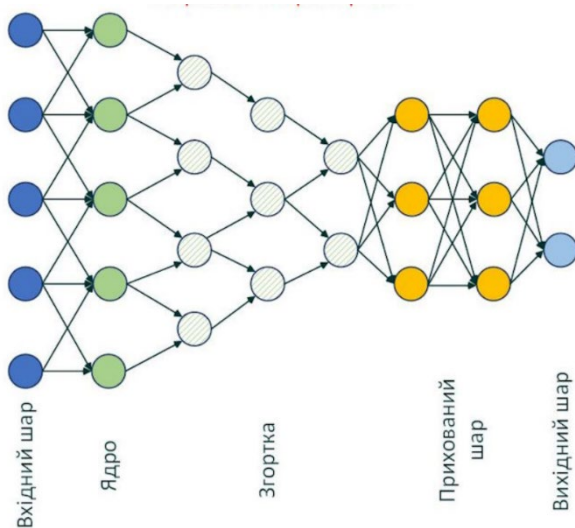


Рисунок 3.3. Схематичне зображення згорткової нейронної мережі

конволюційні шари, – виконують операцію згортки для автоматичного виділення ознак з вхідних даних. Кожен конволюційний шар має набір фільтрів (ядра згортки), які ковзають по всьому зображенню і обчислюють продукт з вхідними даними. Це дозволяє виділяти ознаки, такі як краї, кути, текстури. Переваги даної архітектури: конволюційні шари автоматично виділяють найбільш релевантні ознаки з

вхідних даних, знижуючи необхідність ручного виділення ознак; пулінгові шари дозволяють зменшити розмірність даних, зберігаючи важливу інформацію і зменшуючи обчислювальні витрати (результатом є зменшена версія вхідних даних, де кожне значення представляє максимальне або середнє значення в кожній підобласті даних); CNN можуть бути масштабованими для обробки великих зображень і відео, що робить їх ідеальними для застосувань в комп'ютерному зорі.

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN, рис. 3.4) – це архітектура нейронних мереж, які спеціально розроблені для роботи з послідовними даними, такими як текст, часові ряди, аудіо та відео. Вони можуть запам'ятовувати попередні стани та використовувати їх для прийняття рішень у поточний момент часу, що робить їх потужними для обробки послідовностей та часових залежностей. Приховані стани дозволяють RNN запам'ятовувати інформацію про попередні стани на вході і використовувати її для прийняття рішень у поточний момент часу. Стан прихованих шарів залежить від поточних даних на вході та попереднього стану прихованих шарів. Результати з'являються на кожному кроці послідовності або після обробки всієї послідовності даних у

залежності від задачі. Рекурентні нейронні мережі (RNN) дозволяють моделювати часові залежності даних та використовуються в задачах, де контекст і порядок даних мають важливе значення, таких як машинний переклад,

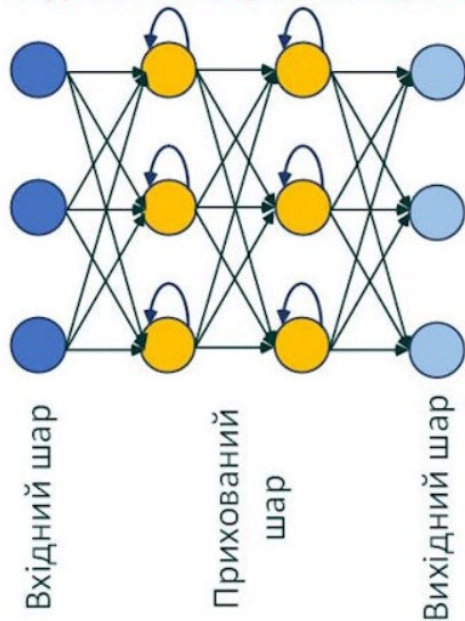


Рисунок 3.4. Схематичне зображення рекурентної нейронної мережі

розпізнавання мовлення та аналіз часових рядів.

Генеративно-змагальні мережі (Generative Adversarial Networks, GAN) є потужними моделями машинного навчання, які складаються з двох нейронних мереж: генератора і дискримінатора. Ці мережі змагаються одна з одною, що дозволяє генератору створювати все більш реалістичні дані. Генератор приймає випадковий шум як вхід і перетворює його на зразки даних, які мають вигляд справжніх даних.

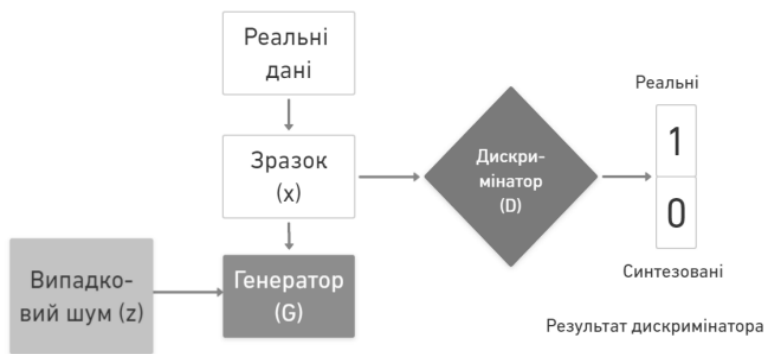


Рисунок 3.5. Структура генеративно-змагальної нейронної мережі [31]

Дискримінатор приймає на вхід дані та визначає, чи вони справжні (з навчального набору) чи створені генератором. Генератор намагається створювати все більш реалістичні дані, щоб “обдурити” дискримінатор. Він оновлює свої ваги на основі

зворотного зв'язку від дискримінатора. Генератор і дискримінатор навчаються одночасно в циклічному процесі, де генератор стає кращим у створенні даних, а дискримінатор стає ефективнішим у розрізненні справжніх і фальшивих даних.

GAN використовуються для створення нових текстів на основі заданого стилю або тематики, для створення реалістичних зображень, які можуть імітувати існуючі фотографії або створювати нові зображення.

3.3. Інструментальні засоби реалізації

В останні роки з'явилося багато програмних засобів і окремих бібліотек для реалізації нейронних мереж, які дозволяють інженерам і дослідникам легко створювати, навчати і розгортати моделі машинного навчання. Деякі з найпопулярніших і широко використовуваних програмних засобів включають TensorFlow, PyTorch, Keras, scikit-learn, Jupyter Notebook, Google Colab, та Microsoft Azure Machine Learning [32, 33].

TensorFlow (Google) – потужна бібліотека з відкритим вихідним кодом для машинного і глибокого навчання. Підтримує розробку як дослідницьких прототипів, так і продукційних систем. За допомогою TensorFlow можна побудувати глибокі нейронні мережі для розпізнавання образів (наприклад, рукописного тексту), рекурентні нейронні мережі для NLP (обробки природних мов). Також тут є модулі для векторизації слів (embedding) та розв'язання диференціальних рівнянь у частинних похідних (PDE). Цей фреймворк має відмінну архітектурну підтримку, що дозволяє проводити обчислення на різних платформах, у тому числі на десктопах, серверах і мобільних пристроях. Основний козир TensorFlow – це абстракції. Вони дозволяють розробникам сфокусуватися на загальній логіці програми, а чи не на дрібних деталях реалізації тих чи інших алгоритмів. Основні можливості: легке створення і навчання моделей нейронних мереж; підтримка як CPU, так і GPU для швидшого навчання; інтеграція з іншими інструментами Google, такими як Colab і TensorBoard для візуалізації процесу навчання.

PyTorch (Facebook AI Research (FAIR)) – гнучка і легка у використанні бібліотека для глибокого навчання, яка набирає популярність серед дослідників і розробників. Основні можливості: динамічне обчислення графів, що дозволяє легко налагоджувати і змінювати моделі; підтримка модулів для обробки зображень, тексту і багато іншого; інтеграція з різними візуалізаційними інструментами, такими як TensorBoardX.

Keras (Франсуа Шолле (François Chollet)) – високорівнева API для глибокого навчання, що працює поверх TensorFlow. Легко вивчати і використовувати для швидкого створення моделей нейронних мереж. У цій бібліотеці реалізовано практично всі автономні модулі нейронної мережі, включаючи оптимізатори, нейронні шари, функції активації шарів, схеми ініціалізації, функції втрат та моделі регуляризації. Це дозволяє будувати нові модулі нейромережі, просто додаючи функції чи класи. І оскільки модель вже визначена у коді, розробнику годі було створювати нею окремі конфігураційні файли. Також Keras можна використовувати під час роботи з згортковими нейронними мережами. У ньому реалізовано алгоритми нормалізації, оптимізації та активації шарів. Основні можливості: спрощує створення і навчання нейронних мереж завдяки зручному синтаксису; підтримка різних бекендів (TensorFlow, Theano, PlaidML); вбудовані моделі для стандартних задач, таких як класифікація зображень та послідовностей.

Scikit-learn (scikit-learn) – бібліотека для машинного навчання мовою Python, яка пропонує інструменти для аналізу даних і побудови моделей. Хоча вона не спеціалізується на глибокому навчанні, може використовуватися для базових задач і інтеграції з іншими інструментами. Основні можливості: підтримка різних алгоритмів машинного навчання, включаючи класифікацію, регресію і кластеризацію; інтеграція з NumPy, SciPy і Matplotlib для аналізу даних та візуалізації.

Jupyter Notebook (Jupyter) – інтерактивне середовище для написання і виконання коду, що підтримує більше 40 мов програмування, включаючи Python. Основні можливості: зручна платформа для інтерактивного аналізу даних і навчання моделей; інтеграція з іншими інструментами для візуалізації даних і результатів моделювання; підтримка віджетів для взаємодії з даними у реальному часі.

Google Colab (Google) – безкоштовне хмарне середовище для розробки і тренування моделей машинного навчання. Базується на Jupyter Notebook і забезпечує доступ до потужних апаратних ресурсів, таких як GPU і TPU. Основні

можливості: легкий доступ до GPU і TPU для швидшого навчання моделей; можливість імпортувати дані з Google Drive і інтегрувати з іншими Google-сервісами; спільна робота над проектами у режимі реального часу.

Microsoft Azure Machine Learning (Microsoft) – хмарна платформа для розробки, навчання і розгортання моделей машинного навчання. Основні можливості: інтеграція з іншими сервісами Microsoft Azure; підтримка автоматичного машинного навчання для швидкої побудови моделей; інструменти для управління експериментами, моніторингу і налагодження моделей.

3.4. Синтез нейронної мережі для вирішення задачі

Синтез оптимальної структури нейронної мережі полягає у визначенні кількості рівнів нейронної мережі, числа шарів та нейронів прихованого шару, виборі методу навчання та способу оцінки результату навчання. Задача синтезу нейронної мережі вирішується у кілька етапів. Критерієм оптимізації структури нейронної мережі вибрано точність класифікації.

На першому етапі оцінювалася потенційна можливість використання нейронних мереж на вирішення задачі класифікації задач для звичайних диференціальних рівнянь. За основу обрано однорівневу двошарову нейронну мережу з прямими зв'язками та навчанням за методом зворотного поширення помилки (рис. 3.1). Класи відповідають об'єктам: “ЗДР 2 порядку однорідне з сталими коефіцієнтами”, ”ЗДР 2 порядку однорідне з змінними коефіцієнтами”, ”ЗДР 2 порядку неоднорідне з сталими коефіцієнтами”, тощо (див. п.2.1).

Нейронна мережа навчалася таким чином, щоб при появі на її вході даних певного класу на відповідному виході був одиничний сигнал. Отримано результати навчання та тестування класифікатора. Передбачено можливість месштабування нейронної мережі для класифікації диференціальних рівнянь інших типів (не включених у класифікатор на початковому етапі).

На другому етапі зроблено припущення про те, що підвищити ймовірність правильної класифікації можна за рахунок введення додаткових рівнів нейронної

мережі, аналогічних попередньому, із загальним входом для всіх рівнів та окремим виходом для кожного з них. На кожному рівні використано двошарову нейронну мережу. Навчання проводилося таким чином, щоб на виході був одиничний сигнал за відповідного вхідного впливу.

Були отримані більш високі (хоч і нижче за необхідний рівень 0,9) оцінки ймовірності правильної класифікації.

Третій етап: розроблено двошарову нейронну мережу для класифікації початкових/граничних умов. Також побудовано нейронну мережу відповідно до послідовної схеми прийняття рішень: на першому рівні приймається рішення про виявлення типу рівняння, на другому – розпізнається тип граничних умов.

Обробка інформації на кожному рівні здійснюється незалежно і одночасно з усіма іншими. Рішення приймається по комбінації вихідних сигналів з кожного рівня. На кожному рівні реалізовано багат шарову нейронну мережу.

На четвертому етапі проводиться оцінка якості навчання нейронної мережі запропонованої архітектури. Зроблено відповідні висновки. Для підвищення ефективності класифікатора було ускладнено структуру багаторівневої нейронної мережі за рахунок введення додаткового прихованого шару на кожному рівні.

Наведемо приклад програмного коду, створеного засобами мови Python з використанням бібліотек tensorflow, keras.

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout

# Приклад даних (спрощений для демонстрації)
# Вхідні дані - це числові представлення рівнянь
# Мітки - це категорії рівнянь (0 - лінійне, 1 - нелінійне, 2 - однорідне, 3 -
неоднорідне)
data = np.array([[0.5, 0.3, 0.2], [0.1, 0.4, 0.5], [0.6, 0.1, 0.3], [0.3, 0.6, 0.1]])
labels = np.array([0, 1, 2, 3])

# Побудова моделі нейронної мережі
model = Sequential([
    Dense(64, input_dim=3, activation='relu'),
```

```

        Dropout(0.5),
        Dense(32, activation='relu'),
        Dense(4, activation='softmax')
    ])

    # Компіляція моделі
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])

    # Навчання моделі
    model.fit(data, labels, epochs=100, batch_size=4, verbose=1)

    # Оцінка моделі (при наявності тестових даних)
    # test_data = np.array([...])
    # test_labels = np.array([...])
    # model.evaluate(test_data, test_labels)

    # Приклади використання навченої моделі для класифікації нових рівнянь
    new_data = np.array([[0.4, 0.2, 0.4]])
    predictions = model.predict(new_data)
    predicted_class = np.argmax(predictions)

    print(f"Предсказаний клас: {predicted_class}")

```

Роботи по удосконаленню нейронної мережі (розширенню структури), навчанню (відповідному розширенню навчальних вибірок), по оцінці результативності алгоритмів продовжуються.

ВИСНОВКИ

Перспективними шляхами підвищення ефективності обчислювальних алгоритмів розв’язання різноманітних науково-технічних задач на даному етапі розвитку науки і техніки є залучення символічних перетворень і інтелектуального аналізу. Вирішення математичних задач методами символічних обчислень і використання штучного інтелекту стало можливим у зв’язку із суттєвим прогресом у розвитку відповідних програмних і технічних засобів.

Ефективними алгоритмами розв’язання складних задач математичної фізики (зокрема нелінійних) є ті, які ґрунтуються на методах теорії збурень асимптотичного аналізу. Такі алгоритми потребують автоматизації відповідних процесів аналізу і синтезу, які є досить трудомісткими.

У роботі запропоновано концепцію розробки штучних нейронних мереж ідентифікації задач Коші/крайових задач для звичайних диференціальних рівнянь. Розглянуто відповідну джерельну базу, методи класифікації (ідентифікації) об’єктів, типи штучних нейронних мереж, інструментальні засоби їх реалізації.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Smith D. R. Singular-Perturbation Theory. An Introduction with Applications / D. R. Smith. – Cambridge: Cambridge Univ. Press, 1985. – 520 p. URL: <https://books.google.com.ua/books?id=cEszbdam0zwC&printsec=frontcover&hl=ru#v=onepage&q&f=false>
2. Бомба А. Я. Методи теорії збурень прогнозування процесів тепломасоперенесення в пористих та мікропористих середовищах / А. Я. Бомба, І. М. Присяжнюк, О. В. Присяжнюк. – Рівне: О.Зень, 2017 – 291 с.
3. Vasil'eva A. B. The Boundary Function Method for Singular Perturbation Problems / A. B. Vasil'eva, V. F. Butusov, L. V. Kalachev. – SIAM, Philadelphia, 1995. Doi: <https://doi.org/10.1137/1.9781611970784>
4. Bomba A. Analysis of Nonlinear Processes in the P-I-N Diodes Plasma by the Perturbation Theory Methods / A. Bomba A., I. Moroz // ACIT-2023 Conference Proceedings, Wroclaw, Poland. – P. 117-120.
5. Бомба А. Я. Системний аналіз в математичному та інформаційному моделюванні напівпровідникових р-і-п-структур / А. Я. Бомба, І. П. Мороз, З. С. Лістєв // VIII Міжнародна науково-технічна конференція «Комп'ютерне моделювання та оптимізація складних систем», 1-3 листопада 2023, Запоріжжя. – С. 32-33.
6. Самолук В. П. Системи автоматизованого проектування напівпровідникових структур / В. П. Самолук, І. П. Мороз // XVII Всеукраїнської науково-практичної конференції "Інформаційні технології у професійній діяльності», 5 листопада 2024, Рівне. – С. 235-239.
7. Bomba A. Development of a Software Architecture Concept for Solving Singularly Perturbed Problems in Semiconductor Electronics / A. Bomba, I. Moroz, Z. Listjev // VII Міжнародній науково-практичній конференції «Моделювання, керування та інформаційні технології (MCIT–2024)», 7–9 листопада 2024, Рівне. – С. 167-168.

8. Таранчук В. Б. Основные функции систем компьютерной алгебры / В. Б. Таранчук. – Минск: БГУ, 2013. <http://elib.bsu.by/handle/123456789/46210>.
9. Попов Б. О. Розв'язування математичних задач у системі комп'ютерної алгебри Maple V / Б. О. Попов. – Київ: ViP, 2001. – 312 с. – ISBN 966-7897-03-6.
10. Comparison of computer algebra systems. [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/List_of_computer_algebra_systems.
11. Хіміч О.М. Інтелектуальна система комп'ютерної математики INPARSOLVER / О.М. Хіміч, Т.В. Чистякова, В.А. Сидорук, П.С. Єршов // Штучний інтелект, - 2020, - № 4, - С. 60-71.
12. Сурмін Ю. П. Теорія систем та системний аналіз: Навч. Посібник / . Ю. П. Сурмін – К.: МАУП, 2003. – 368 с.
13. Процеров Ю. С. Асимптотичні методи в аналізі : навч.-метод. посіб. / Ю. С. Процеров. – Одеса : Одес. нац. ун-т ім. І. І. Мечникова, 2024. – 93 с. ISBN 978-617-689-568-8.
14. Андрианов И. Методы асимптотического анализа и синтеза в нелинейной динамике и механике деформируемого твердого тела / И. Андрианов, Я. Аврейцевич. – Ижевск : Институт компьютерных исследований, 2013. – 274 с. ISBN 978-5-4344-0116-6.
15. Miller P. D. Applied Asymptotic Analysis / P. D. Miller. – Providence: American Mathematical Soc., 2006 – 467 p.
16. Bomba A. Ya. The Optimization of the Shape and Size of the Injection Contacts of the Integrated P-I-N-structures on the Base of Using the Conformal Mapping Method / A.Ya. Bomba, I.P. Moroz and M.V. Bojchura // Radio Electronics, Computer Science, Control. – 2021. – no. 1. – P. 14-27.
17. Belyanin M.P. On the Asymptotic Solution of One P-N-junction Model / M.P. Belyanin // U.S.S.R. Comput. Math. Math. Phys. – 1986. – vol. 26 (2). – P. 306-311. URL: <http://www.mathnet.ru/links/f91b663c991258d0733ea2d02cf7f68a/zvmmf9248.pdf>
18. Vasil'eva A.B. Singularly perturbed systems in the theory of semiconductors / A.B. Vasil'eva, V.G. Stel'makh // U.S.S.R. Comput. Math. Math. Phys. – 1977. – vol.

17(2). – Р. 339-348. URL: <http://www.mathnet.ru/links/5d86b56f4673e4843abf1810d61ba4e8/zvmmf6023.pdf>

19. Задачин В. М. Чисельні методи : навчальний посібник / В. М. Задачин, І. Г. Конюшенко. – Х.: Вид. ХНЕУ ім. С. Кузнеця, 2014. – 180 с.

20. Кононова К. Ю. Машинне навчання: методи та моделі: підручник / К. Ю. Кононова. – Харків: ХНУ імені В. Н. Каразіна, 2020. – 301 с.

21. Руденко О. Г. Штучні нейронні мережі: навч. Посібник / О. Г. Руденко, Є. В. Бодянский. – Харків: ТОВ «Компанія СМІТ», 2006. – 404 с.

22. Каллан Р. Основные концепции нейронных сетей / Р. Каллан. – М.: Издательский дом «Вильямс», 2001. – 288 с.

23. Добротвор, І. Г. Системний аналіз: навч. посіб. / І. Г. Добротвор, А. О. Саченко, Л. М. Буяк. – Тернопіль: ТНЕУ, 2019. – 170 с.

24. Ляшко І.І. Диференціальні рівняння. / І. І. Ляшко, О. К. Боярчук, Я. Г. Гай, О. Ф. Калайда. – К.: Вища школа, 1981. – 504 с.

25. Камке Э. Справочник по обыкновенным дифференциальным уравнениям / Э. Камке. – М.: Наука, 1971. – 576 с.

26. Кутковецький В. Я. Розпізнавання образів : навчальний посібник / В. Я. Кутковецький. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2017. – 420 с.

27. Краснопопсовський А. С. Класифікаційний аналіз даних : навчальний посібник / А. С. Краснопопсовський. – Суми: Видавництво СумДУ, 2002. – 159 с.

28. Кононюк А. Е. Общая теория распознавания. – К.: "Освіта України", 2012. – 584 с.

29. Форсайт Д. А. Computer Vision: A Modern Approach Компьютерное зрение. Современный подход / Дэвид А. Форсайт, Джин Понс. – М.: «Вильямс», 2004. – 928 с. – ISBN 0-13-085198-1.

30. Круглов В. В. Искусственные нейронные сети. Теория и практика / В. В. Круглов, В. В. Борисов. – М.: Горячая Линия–Телеком, 2001. – 382 с.

31. Striuk O. METHODS OF APPLIED UTILIZATION OF GENERATIVE ADVERSARIAL NETWORKS IN GRAPHIC DATA PROCESSING / O. Striuk, Y.

Kondratenko // Artificial Intelligence. – 2023. – № 3. – Р. 154-161.
<https://doi.org/10.15407/jai2023.03.154>

32. Keras: бібліотека глибокого навчання на Python. Ви щойно відкрили для себе Keras. [Електронний ресурс] URL: <https://ru-keras.com/home/>

33. Welcome to the TensorFlow Forum [Електронний ресурс] Режим доступу: <https://discuss.tensorflow.org/c/keras/16/none>

ДОДАТКИ

Додаток А. Сертифікат учасника конференції

