

## Пояснювальна записка


до кваліфікаційної роботи  
магістра

на тему: «Дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проектами»

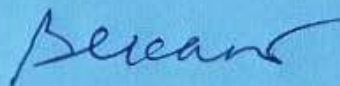
за освітньою програмою **12 Інженерія програмного забезпечення**

зі спеціальності: **121 Інженерія програмного забезпечення**

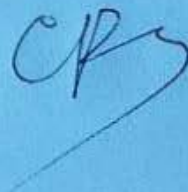
Виконав: студент групи ПЗ2222:

 / Ілля ДУДНИК /

Керівник:

 / Владислав СКАЛОЗУБ /

Нормоконтролер:

 / Світлана ВОЛКОВА /

Засвідчую, що у цій роботі немає  
запозичень з праць інших авторів  
без відповідних посилань.

Студент Дудник І.П.



Ministry of Education and Science of Ukraine  
Ukrainian State University of Science and Technologies

Faculty Computer technologies and systems  
Department Computer information technology

**Explanatory Note**  
to Master's Thesis

on the topic: «Supplementation and modeling of software engineering  
management procedures in project management»  
according to educational curriculum **12 software engineering**  
in the Speciality: **121 software engineering**

Done by the student of the group  
PZ2222:

/ Illia DUDNYK /

Scientific Supervisor:

/ Vladyslav SKALOZUB /

Normative controller:

/ Svitlana VOLKOVA /

Міністерство освіти і науки України  
Український державний університет науки і технологій

Факультет: Комп'ютерних технологій і систем

Кафедра: Комп'ютерні інформаційні технології

Рівень вищої освіти: магістр

Освітня програма: Інженерія програмного забезпечення

Спеціальність: Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ  
Завідувач кафедри \_\_\_\_\_ КІТ  
\_\_\_\_\_ Володимир ГОРЯЧКІН  
\_\_\_\_\_ січень 2024 р.

### ЗАВДАННЯ

На кваліфікаційну роботу Магістра  
студенту Дуднику Іллі Петровичу

1. Тема дипломної роботи: «Дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проектами»

Керівник роботи: Скалозуб Владислав Васильович

затверджені наказом 1196 ст від 05.12. 2022 року

2. Строк подання студентом роботи 11 січня 2024 року

3. Вихідні дані до дипломної роботи: Вимоги стандартів SWBOK, PMBOK до складу та завдань процедур менеджменту програмної інженерії при управлінні проектами, приклади постановок завдань управління проектами сфери програмної інженерії, формулювання завдань призначення кандидатів на проєкт або на ролі виконавців проєктів, моделі та методи формування інтелектуальних процедур автоматизації призначень, процедури нейронних мереж Хеммінга для класифікації за неповних та збурених даних,

приклади застосування моделей нейронних мереж Хеммінга для класифікації на основі показників ступеня належності нечітких величин, а також коефіцієнтів упевненості.

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Аналітична частина: огляд досліджень щодо моделей, методів та прикладів реалізації завдань сфери менеджменту програмної інженерії при управлінні проєктами, в тому числі з використанням інтелектуальних процедур та відомих програмних засобів, постановки нових завдань менеджменту програмної інженерії при управлінні розподілом ресурсів при невизначеності даних, формування завдань класифікації призначень з неточно визначеними ознаками на основі модифікованих мереж Хеммінга.

4.2 Основна частина: розробка математичної моделі завдань класифікації призначень з неточно визначеними ознаками на основі модифікованих мереж Хеммінга, а також процедур класифікації при встановлених вимогах до достовірності результатів на основі методів граничних спрощень і статистики «каппа Коена», результати чисельних експериментальних досліджень алгоритмів і процедур класифікації на основі методів редукції розмірності та статистичної узгодженості, розробка алгоритмів і програмних засобів класифікації об'єктів із заданим рівнем достовірності при неточно визначених характеристиках, проведення числових експериментів з аналізу ефективності запропонованих алгоритмів, розроблення програмних засобів для класифікації об'єктів при встановленому рівні достовірності, а також при неточно визначених характеристиках завдань вибору, результати економічного аналізу програмних засобів, технічна документація.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): розробка презентації щодо результатів дослідження моделей та алгоритмів класифікації завдань призначень при встановленому

рівні достовірності, постановки нових завдань призначення та інтелектуальні процедури Хеммінга при їх реалізації, моделі перетворення неточно визначених характеристик даних «кандидатів-виконавців» у показники шаблонів мережі Хеммінга, результати чисельних експериментальних досліджень алгоритмів і процедур класифікації із встановленому рівні достовірності, програмний комплекс реалізації удосконалених процедур класифікації при неповних і неточно визначених та збурених даних, відео-демонстрація функціонування програмного комплексу.

#### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів	Примітка
1	Вступ	15.01.2023 – 20.01.2023	
2	Постановка задачі, технічне завдання	15.02.2023 – 15.05.2023	30%
3	Розробка інструментальних засобів дослідження	15.05.2023 – 15.07.2023	
4	Виконання досліджень	15.07.2023 – 15.09.2023	60%
5	Оформлення тез доповідей	15.11.2023 – 23.11.2023	
6	Оформлення пояснювальної записки	15.01.2023 – 31.12.2023	
7	Розробка демонстраційних матеріалів	03.01.2024 – 15.01.2024	100%
8	Подання кваліфікаційної роботи до кафедри	01.01.2024	
9	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	25.01.2024	

Студент \_\_\_\_\_ / Ілля ДУДНИК /

Керівник роботи \_\_\_\_\_ / Владислав Васильович

/

## РЕФЕРАТ

Об'єктом дослідження є інтелектуальні програмні засоби моделювання процедур менеджменту програмної інженерії при управлінні проєктами.

Предметом дослідження є математичні моделі та алгоритми процедур менеджменту програмної інженерії при управлінні проєктами на основі інтелектуальних процедур асоціативної пам'яті мережі Хеммінга при не точно визначених характеристиках даних, а також на основі процедур редукції математичних моделей і статистики «каппа Коена».

Метою роботи являється є розвиток постановок завдань та удосконалення математичних моделей для завдань менеджменту програмної інженерії при управлінні проєктами, При цьому також передбачається реалізація завдань відтворення достовірних багатопараметричних моделей класифікації (ДБМК) при невизначених даних на основі застосування процедур редукції (метод граничних спрощень) та каппа статистики. Математичними моделями ДБМК являються структури параметрів (кількість і склад), тип даних і множина шаблонів, які складають базу даних/знань досліджуваної предметної області.

Методи дослідження: методи порівняльного аналізу, статистики, класифікації багатопараметричних об'єктів, методи нечіткого моделювання та методи експертних систем, методи експериментальних досліджень та комп'ютерне моделювання. Методи програмної інженерії використовувалися для проектування та розробки програми.

Результати та їх новизна: виконано аналіз моделей та процедур широкого кола завдань менеджменту програмної інженерії при управлінні проєктами, запропоновано нові постановки завдань призначення на посаду (вибору кандидата), класифікації за нечіткими даними та ознаками у формі коефіцієнтів упевненості, для реалізації завдань класифікації при невизначених даних модифіковані моделі асоціативної пам'яті Хеммінга, а

для забезпечення результатів класифікації з встановленими ймовірнісними показниками розроблено процедуру редукції розмірності математичної моделі з використанням статистики «капа Коена», які застосовуються для завдань класифікації за неточно визначеними даними, проведений широкий числовий експеримент підтвердив достовірність та ефективність запропонованих моделей і методів. Отримано програмну реалізацію розроблених моделей класифікації для менеджменту програмної інженерії при управлінні проєктами з неточно визначеними параметрами.

Розрахунково-пояснювальна записка складається зі вступу, 5 розділів, висновків, бібліографічного списку та додатків.

Вступ – представляє сутність завдань моделювання процедур менеджменту програмної інженерії при управлінні проєктами при неточно визначених характеристик даних. Визначає актуальність завдань досліджень та розробок (2 сторінки).

Перший розділ – містить опис матеріалів наукових літературних джерел щодо сутності та результатів завдань моделювання процедур менеджменту програмної інженерії при управлінні проєктами при неточно визначених характеристик даних, методів і процедур нейронних мереж Хеммінга, особливостей даних необхідних для процедур класифікації щодо призначення «кандидатів на посаду» (14 сторінок).

Другий розділ – містить результати розвитку постановок завдань менеджменту програмної інженерії при управлінні проєктами на основі інтелектуальних процедур при не точно визначених характеристиках даних, моделі завдань призначення за умов неточно визначених даних, результати модифікування моделей нейронних мереж Хеммінга, нечіткі та форми представлення неточно визначених даних у вигляді коефіцієнтів упевненості, процедуру редукції математичних моделей завдань класифікації на основі статистики «каппа Коена», а також результати чисельних досліджень

запропонованих алгоритмів та процедур модифікації нейронної мережі Хеммінга. (15 сторінок).

Третій розділ – предствіляє формування структури програмного комплексу щодо завдань менеджменту програмної інженерії при управлінні проєктами на основі моделей класифікації за неточно визначеними даними, результати проектування і розробки програми (19 сторінок).

Четвертий розділ – містить результати експериментальних досліджень можливостей запропонованих процедур програмного комплексу щодо завдань менеджменту програмної інженерії при управлінні проєктами для неточно визначених даних, які підтвердили достовірність та ефективність результатів досліджень та розробок. (17 сторінок).

Висновки Складаються з 1 сторінки. Додатки – технічне завдання і робочий проєкт. Таблиць – 9, рисунків – 45 , бібліографія – 39.

Ключові слова: менеджмент при управлінні проєктами, інтелектуальні процедури класифікації, модифіковані нейронні мережі Хеммінга, нечітка модель даних, модель даних типу коефіцієнтів впевненості, процедура редукції, статистика каппа Коена, багатовимірна класифікація, програмне забезпечення. Пояснювальна записка до кваліфікаційної роботи магістра: (рівень освіти) 139с., 45 рис., 9 табл., 4 додатки, 39 джерел.

## Зміст

Вступ.....	12
1 ДОСЛІДЖЕННЯ ЗАВДАНЬ МЕНЕДЖМЕНТУ ПРИ УПРАВЛІННІ ПРОЄКТАМИ, ПРОЦЕДУРИ ТА ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ ЗАВДАНЬ З НЕТОЧНО ВИЗНАЧЕНИМИ ДАНИМИ	<b>Помилка! Закладку не визначено.</b>
1.1 Аналіз результатів моделювання завдань менеджменту при управлінні проєктами з урахуванням невизначеності даних аналізу	<b>Помилка! Закладку не визначено.</b>
1.1.1 Моделі та методи завдань менеджменту при управлінні проєктами. Актуальність дослідження .....	<b>Помилка! Закладку не визначено.</b> 5
1.1.2 Аналіз математичних моделей та інтелектуальних засобів менеджменту при управлінні проєктами з урахуванням невизначеності даних.....	<b>Помилка! Закладку не визначено.</b> 6
1.1.3 Огляд програмних аналогів щодо менеджменту управління проєктами за умов невизначеності даних	<b>Помилка! Закладку не визначено.</b> 9
1.2 Характеристика сфери використання процедур менеджменту управління проєктами за умов невизначеності даних	<b>Помилка! Закладку не визначено.</b>
1.3 Постановка завдань менеджменту управління проєктами за умов невизначеності даних на основі модифікованих процедур асоціативної пам'яті та редукції математичних моделей класифікації .....	<b>Помилка! Закладку не визначено.</b> 3
Висновки .....	<b>Помилка! Закладку не визначено.</b> 8
2 РОЗРОБКА ТА ДОСЛІДЖЕННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА ПРОЦЕДУР МЕНЕДЖМЕНТУ УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ДАНИХ НА ОСНОВІ МОДИФІКОВАНИХ МЕРЕЖ ХЕММІНГА ТА ПРОЦЕДУР РЕДУКЦІЇ.....	<b>Помилка! Закладку не визначено.</b> 9
2.1 Представлення вихідних даних для завдання визначення кандидата на місце.....	<b>Помилка! Закладку не визначено.</b> 9

2.2	Структури шаблонів завдань класифікації. ....	31
2.3	Схема процедури формування шаблонів класифікації зі встановленим рівнем достовірності результатів (Пкап). <b>Помилка! Закладку не в</b>	
2.4	Моделювання та Класифікація в Умовах Невизначеності для Оптимізації Технологічних Процесів <b>Помилка! Закладку не визначено.</b>	8
	Висновки .....	<b>Помилка! Закладку не визначено.</b> 4
3	ФОРМУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ЗАВДАНЬ МЕНЕДЖМЕНТУ УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНОЇ ІНЖЕНПЕРІЇ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ДАНИХ <b>Помилка! Закладку не в</b>	
3.1	Постановка та формалізація завдань дослідження інтелектуальних процедур нейронної мережі Хеммінга з неточно визначеними даними .....	<b>Помилка! Закладку не визначено.</b> 5
3.2	Аналіз Конкуруючих Варіантів та Методів Оцінки Моделей Класифікації.....	<b>Помилка! Закладку не визначено.</b> 8
3.3	Програмне Забезпечення та Процедури Редукції Розмірності Моделей для Управління Проектами в Області Програмної Інженерії.....	<b>Помилка! Закладку не визначено.</b>
	Висновки .....	<b>Помилка! Закладку не визначено.</b> 4
4	ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСОБІВ МЕНЕДЖМЕНТУ УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНОЇ ІНЖЕНПЕРІЇ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ДАНИХ <b>Помилка! Закладку не визначено.</b>	5
4.1	Дослідження процедур нечіткої класифікації та редукції моделей даних об'єктів. ....	<b>Помилка! Закладку не визначено.</b> 5
4.2	Дослідження процедур класифікації на основі даних достовірності $CF(X)$ . ....	<b>Помилка! Закладку не визначено.</b> 9
4.3	Процедура Каппа Коена і редукції ...	<b>Помилка! Закладку не визначено.</b> 3
4.4	Дослідження достовірності процедур нечіткої класифікації щодо завдань мнеджменту управління проектами за умов невизначеності . <b>Помилка! З</b>	
	Висновки .....	<b>Помилка! Закладку не визначено.</b> 2

Список ВИКОРИСТАНИХ ДЖЕРЕЛ .....	<b>Помилка! Закладку не визначено.</b>	<b>5</b>
Додатки .....	<b>Помилка! Закладку не визначено.</b>	<b>9</b>
ДОДАТОК А .....	<b>Помилка! Закладку не визначено.</b>	<b>9</b>
ДОДАТОК Б .....	<b>Помилка! Закладку не визначено.</b>	<b>7</b>
ДОДАТОК В.....	<b>Помилка! Закладку не визначено.</b>	<b>6</b>
ДОДАТОК Г .....	<b>Помилка! Закладку не визначено.</b>	<b>3</b>

## ВСТУП

Управління проектами у сфері програмної інженерії вимагає систематичного підходу та ефективного використання ресурсів для досягнення поставлених цілей. У зв'язку зі зростанням складності та обсягу проектів програмної інженерії, важливо мати належне розуміння процедур менеджменту, які забезпечують успішне виконання проектів. Ця стаття присвячена дослідженню та моделюванню процедур менеджменту програмної інженерії при управлінні проектами.

- Розуміння процесу програмного проектування

Перед початком будь-якого проекту програмної інженерії необхідно мати чітке розуміння процесу програмного проектування. Цей процес включає фази, такі як аналіз вимог, проектування архітектури, розробка, тестування та впровадження. Крім того, необхідно визначити завдання та цілі проекту, розподілити ролі та відповідальність між членами команди.

- Вибір процесу розробки програмного забезпечення

Управління проектами в програмній інженерії вимагає вибору оптимального процесу розробки програмного забезпечення. Існують різні моделі процесів розробки, такі як водопадна модель, ітеративна модель, спіральна модель та Agile-методології, такі як Scrum та Kanban. Вибір процесу залежить від характеру проекту, розміру команди, термінів та вимог замовника. Моделювання різних процесів дозволяє оцінити їх переваги та недоліки і обрати найбільш підходяще.

- Управління ризиками

Ефективне управління ризиками є ключовим аспектом успішного проекту програмної інженерії. Дослідження та моделювання процедур управління ризиками допомагають ідентифікувати потенційні ризики, оцінити їх вплив та імплементувати стратегії зниження ризиків. Це може

включати планування запобіжних заходів, розробку плану реагування на ризики та використання інструментів моніторингу та контролю.

- Використання проектного менеджменту

Проектний менеджмент є невід'ємною частиною управління проектами в програмній інженерії. Дослідження та моделювання процедур проектного менеджменту допомагають встановити ефективну систему планування, ресурсного управління, моніторингу прогресу та контролю якості. Застосування методів та інструментів, таких як критерії SMART (специфічні, вимірювані, досяжні, реалістичні, обмежені за часом), діаграми Ганта, методи оцінки ризиків та оцінка продуктивності команди, сприяє забезпеченню успіху проекту.

- Впровадження комунікаційних процедур

Ефективна комунікація є важливою складовою процедур менеджменту програмної інженерії при управлінні проектами. Дослідження та моделювання комунікаційних процедур допомагають визначити канали комунікації, встановити регулярність та структуру звітності, забезпечити ефективну обмін інформацією та залучення зацікавлених сторін. Використання комунікаційних інструментів, таких як електронна пошта, веб-конференції, спільні робочі простори та професійні мережі, сприяє покращенню комунікації всередині команди та зі стейкхолдерами.

Дослідження та моделювання процедур менеджменту програмної інженерії при управлінні проектами є важливим кроком у забезпеченні успіху проектів. Це дозволяє розробити чіткі та ефективні процеси, визначити ролі та відповідальність, управляти ризиками та забезпечити ефективну комунікацію. Шляхом дослідження та моделювання процедур менеджменту програмної інженерії, організації забезпечується можливість уникнути проблем та збільшити ймовірність успіху виконання проектів програмної інженерії.

# 1 ДОСЛІДЖЕННЯ ЗАВДАНЬ МЕНЕДЖМЕНТУ ПРИ УПРАВЛІННІ ПРОЄКТАМИ, ПРОЦЕДУРИ ТА ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ ЗАВДАНЬ З НЕТОЧНО ВИЗНАЧЕНИМИ ДАНИМИ

## 1.1 Аналіз результатів моделювання завдань менеджменту при управлінні проєктами з урахуванням невизначеності даних аналізу

Аналіз результатів моделювання завдань менеджменту при управлінні проєктами з урахуванням невизначеності даних є критичним для визначення ефективності стратегій та процедур, використовуваних під час виконання проєктів в умовах невизначеності. Розглянемо ключові аспекти аналізу результатів:

**Ефективність Стратегій:** Проведений аналіз дозволяє оцінити ефективність обраного стратегічного підходу до управління невизначеністю. Це включає визначення, наскільки обрані стратегії дозволяють ефективно вирішувати проблеми, пов'язані з невизначеністю та змінами в умовах проєкту.

**Управління Ризиками:** Аналіз даних дозволяє визначити ефективність системи управління ризиками. Оцінюється, наскільки успішно ідентифікуються та обробляються ризики, які виникають в умовах невизначеності, і як це впливає на результативність проєкту.

**Гнучкість та Адаптабельність:** Аналізується готовність системи управління адаптуватися до змін. Перевіряється, наскільки ефективно використовуються ітераційні підходи та гнучкість в управлінні для адаптації до невизначеності.

**Використання Технологій:** Вивчається ефективність використання сучасних інструментів та технологій для збору, аналізу та візуалізації даних, спрямованих на управління невизначеністю. Оцінюється, наскільки ці технології полегшують процеси управління.

Залучення Команди: Аналізується, наскільки система управління сприяє активному залученню команди та як це впливає на якість прийняття рішень в умовах невизначеності.

Стратегія Заходів у Невизначеності: Вивчається ефективність розроблених стратегій та планів дій для реагування на невизначеність. Оцінюється здатність системи управління швидко реагувати та адаптуватися до змін у зовнішньому середовищі.

Відзначення Успіхів та Помилки: Проведений аналіз дозволяє виокремити успіхи та помилки в управлінні невизначеністю та використовуваних стратегіях, надаючи можливість вдосконалення підходів до подібних завдань у майбутньому.

Загалом, результати аналізу надають цінний внесок у подальшу оптимізацію процедур та стратегій управління проектами в умовах невизначеності, сприяючи забезпеченню ефективності та успішності виконання проєктів.

### **1.1.1 Моделі та методи завдань менеджменту при управлінні проєктами. Актуальність дослідження**

Актуальність роботи: Актуальність проведення дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проєктами визначається сучасними тенденціями в розробці програмного забезпечення та управлінні його життєвим циклом. Швидкі технологічні зміни, зростання складності програмних продуктів та висока конкуренція висувають нові виклики перед менеджерами програмних проєктів. Досягнення ефективного управління вимагає розробки нових методів та моделей, які забезпечать більш точне та оптимальне вирішення завдань у сфері управління проєктами програмної інженерії.

Об'єкт та предмет дослідження: Об'єкт дослідження: Об'єктом даної роботи є процедури менеджменту, застосовані в галузі програмної інженерії

при управлінні програмними проєктами. Предмет дослідження: Предметом дослідження є методи та моделі, використовувані для управління людським потенціалом, вартістю, навчанням та іншими аспектами, що впливають на результативність та успішність програмних проєктів.

Мета роботи: Метою роботи є розвиток постановок завдань та удосконалення математичних моделей оптимізації потоків замовлень сервісних систем на основі інтелектуальних процедур Хеммінга при не точно визначених характеристиках даних.

Метою даної роботи є проведення наукового дослідження та моделювання процедур менеджменту в області програмної інженерії при управлінні проєктами з метою:

Аналізу актуальних проблем та викликів у сфері менеджменту програмних проєктів: Розкриття та визначення сучасних викликів, що виникають в процесі управління проєктами програмної інженерії, зокрема, у встановленні зв'язку між процесом підтримки та управління та ключовими аспектами, такими як вартість, людські ресурси, навчання та інструменти.

Розробки нових методів та моделей для ефективного управління проєктами: Створення нових підходів, моделей та методів для оптимізації управління процесами програмної інженерії, зокрема, у вирішенні завдань, пов'язаних з призначенням на посаду та розподілом персоналу між проєктами.

### **1.1.2 Аналіз математичних моделей та інтелектуальних засобів менеджменту при управління проєктами з урахуванням невизначеності даних**

Одне із головних завдань сфери менеджменту програмної інженерії (ПІ) при управлінні проєктами полягає в тому, щоб встановити зв'язок між діяльністю процесу підтримки та управління і питаннями вартості, людських ресурсів, навчання та інструментів ін. [1, 2]. Управління і

використання людського потенціалу представляють відповідальну і важливу складову завдань менеджменту ПЗ [1]. Серед цих завдань виділяється окремо завдання щодо призначення на посаду або розподілу персоналу між проєктами, коли необхідно враховують можливості та досвід «кандидатів», претендентів. Для обґрунтованого вирішення такого типу завдань планування (на основі математичних моделей і алгоритмів), з урахуванням кількості, достовірності та форми надання вихідної інформації про кандидатів, застосовують значну кількість підходів і процедур [1]. Такі завдання також можливо звести до моделей завдань багатоапараметричної класифікації. В цих моделях кожний «кандидат» на відповідне місце характеризується набором параметрів (як правило, однаковим для всіх), які приймають певні значення відповідно властивостям кандидатів. Місце призначення, роль або проєкт, до якого обирається «кандидат», також кодується тим самими наборами параметрів-ознак зі значеннями оцінок вимог до виконавця («еталон»). За рахунок певної процедури класифікації необхідно серед учасників відбору встановити «кращого» із претендентів, оцінки показників якого найбільше відповідають вимогам.

Наведене таке змістовне завдання широко відоме, має багато моделей і процедур реалізації, в залежності від типу та ознак параметрів, способів їх отримання а оцінювання, а також ступеня достовірності даних про «кандидатів» і вимог «еталону». Особливість завдання щодо призначення, яке досліджене у цій роботі, складається із наступного.

6. Вважається, що менеджментом реалізується процедура вибору «кандидата» на роль або до певного проєкту ПЗ на основі даних його портфоліо (переліку виконаних робіт у певних розробках програм). При цьому приймається до уваги, що кожний з кандидатів  $K_i$  має набір реалізованих завдань  $Z_i$  портфоліо  $P_i$ , які описані без попередніх вимог до структури опису  $Z_i$ . Також параметри та оцінки за  $Z_i$  можуть відрізнятися для різних  $P_i$  як за кількістю, так і за складом ознак та формою

представлення оцінок, у тому числі поданих у формі неточних показників ін. Таким чином, маємо завдання класифікації  $K_i$  за неповними та неточно визначеними і збуреними даними. При чому остаточну структуру і оцінки щодо представлення  $Z_{ik}$  визначає менеджмент.

7. Для реалізації завдання класифікації при неоднорідних, неточних та неповних даних в роботі застосовуються модифіковані мережі Хеммінга, які призначені для класифікації на основі нечітких даних та даних у формі коефіцієнтів упевненості  $CF(A)$ .

8. Розробляються дві процедури формування шаблонів класифікації, тобто моделей описання властивостей «кандидатів»  $K_i$  і вимог до моделі виконавця, тобто «еталону».

- В першій процедурі шаблони представляють певні узагальнені оцінки складових  $Z_{ik}$  набору  $P_i$ .

- В другій процедурі в якості шаблонів використовуються закодовані структури окремих  $Z_{ik}$ . Виконується порівняльний аналіз щодо ефективності наведених процедур та форм утворення шаблонів завдань класифікації на основі модифікованої моделі Хеммінга.

9. В роботі запропоновано нову процедуру формування шаблонів, моделей для класифікації «кандидатів»  $K_i$ , які забезпечують рішення щодо визначення результату з встановленими ймовірнісними вимогами. Для цього використовуються методи спрощень (МСП), та статистика «каппа Коена»,  $K$ -каппа. Розроблені на основі МСП та  $K$ -каппа процедури забезпечують формування шаблонів класифікації «кандидатів»  $K_i$  з числом параметрів, що не перевищує граничну межу  $n_0$ . Реалізація такої процедури для завдань призначення в літературі нами не знайдена.

10. Для реалізації завдань призначення на основі  $P_i$  та моделювання процедур менеджменту програмної інженерії при управлінні проєктами розроблене відповідне програмне забезпечення.

### **1.1.3 Огляд програмних аналогів щодо менеджменту управління проектами за умов невизначеності даних**

Історія розвитку управління програмними проектами є важливим аспектом у вивченні та розумінні еволюції методів та процедур, що використовуються в програмній інженерії. Низка ключових подій та етапів формування управлінської парадигми у цій галузі визначає сучасні стандарти та методології.

#### **Початковий етап (1950-1970)**

На початку розвитку програмної інженерії, коли виникли перші електронні обчислювальні машини, управління проектами було поглиблено в технічних аспектах програмування. Проекти виконувались індивідуально, а в основі управління була технічна ефективність.

#### **Етап методологій (1970-1980)**

В середині 1970-х років виникли перші методології управління проектами, такі як методологія каскадного виконання проектів. З'явилися стандарти і моделі, наприклад, Capability Maturity Model (СММ), що спрямовані на підвищення якості та ефективності розробки ПЗ.

#### **Зростання гнучкості (1980-1990)**

У 1980-1990 роках зародилися гнучкі методології, такі як Extreme Programming (XP) та Scrum. Основна ідея - адаптація до змін у вимогах і швидка реакція на зміни. Цей період поклав початок широкому використанню принципів гнучкого управління проектами.

#### **Етап сучасності (1990-до сьогодні)**

Сучасний етап управління програмними проектами характеризується використанням Agile-методологій, DevOps, та інших сучасних підходів. Ключовими рисами є ітеративність, активна участь замовників, автоматизація та швидка поставка програмного продукту.

### **Аналіз існуючих рішень**

### Каскадний метод (Waterfall):

Каскадний метод визначає чітку послідовність етапів розробки, де кожен наступний етап починається тільки після завершення попереднього. Його прогнозована структура забезпечує якість, але невгамовна для змін. Такий підхід ефективний для проектів зі статичними вимогами.

Каскадний метод передбачає лінійний послідовний підхід до розробки, де кожен етап проекту завершується перед переходом до наступного. Етапи включають визначення вимог, проектування, реалізацію, тестування, впровадження та підтримку.

#### Переваги:

**Прогнозованість:** Чітка послідовність дозволяє точно прогнозувати терміни та вартість проекту.

**Висока якість:** Кожен етап має свою фазу тестування, що забезпечує високу якість кінцевого продукту.

#### Недоліки:

**Нееластичність:** Зміни у вимогах під час розробки важко впроваджувати.

**Обмежена взаємодія замовника:** Замовник бачить результат лише в кінці проекту.

### Гнучкі методології (Scrum, Kanban):

Гнучкі методології реалізують ітераційний підхід до розробки, де продукт постійно вдосконалюється за участі замовника. Метод Scrum використовує короткі ітерації (спринти), підсилюючи комунікацію в команді, а Kanban наголошує на гнучкості та плавному потоці роботи.

Гнучкі методології базуються на ітераційному та інкрементальному підході до розробки. Scrum спрямований на фіксований часовий період, називаний "спринт", під час якого команда працює над визначеними завданнями. Kanban, натомість, акцентується на візуальному контролі за

процесом і обмеженні робочих завдань, що можуть бути взяті на кожному етапі.

Переваги:

Гнучкість: Здатність адаптуватися до змін у вимогах під час розробки.

Взаємодія з замовником: Замовник бере активну участь, бачачи проміжні результати на кінці кожного спринту.

Недоліки:

Складніше планування: Гнучкі методології можуть вимагати більшого управлінського зусилля при плануванні.

Низька прогнозованість: Важко точно прогнозувати терміни завершення проекту через змінливість вимог.

DevOps:

DevOps поєднує розробку та операції, прискорюючи випуск продукту та автоматизуючи процеси. Впровадження DevOps сприяє підвищенню ефективності, але вимагає культурних змін та навчання нових навичок.

Capability Maturity Model (CMM):

Модель Capability Maturity Model визначає етапи еволюції організації у процесі розробки. Цей підхід допомагає систематизувати та підвищити якість процесів, але вимагає значних зусиль на впровадження.

Agile-моделі з підходом Scrum:

Scrum акцентує на регулярних ітераціях, що включають взаємодію замовника та гнучку реакцію на зміни. Деякі виклики виникають при масштабуванні на великі проекти та управлінні розподіленими командами.

## **1.2 Характеристика сфери використання процедур менеджменту управління проектами за умов невизначеності даних**

У сфері управління проектами, особливо в умовах невизначеності даних, процедури менеджменту грають надзвичайно важливу роль у

забезпеченні ефективності та успішності проєктів. Невизначеність може виникнути з-за багатьох факторів, таких як зміни в умовах ринку, технологічні зрушення, непередбачені ризики чи зміни в вимогах замовника. Тому важливо розглядати процедури управління як стратегічний інструмент для ефективного керування проєктами в умовах невизначеності.

**Стратегічне планування:** Умови невизначеності вимагають від менеджерів проєктів розробки агільних стратегій та гнучких планів. Процедури менеджменту управління проєктами дозволяють оперативно реагувати на зміни в умовах проєкту, переглядати та оновлювати плани, що є ключовим управлінським підходом в умовах невизначеності.

**Гнучкість та Адаптивність:** Використання процедур управління дозволяє забезпечити гнучкість в плануванні та виконанні завдань. Компанії здатні адаптуватися до непередбачуваних сценаріїв та змінювати підходи для досягнення поставлених цілей.

**Ризик-менеджмент:** Процедури управління проєктами акцентують на ефективному виявленні та управлінні ризиками, визначенні стратегії запобігання та плани реагування на невизначені ситуації. Це допомагає знижувати ймовірність виникнення проблем та втрат.

**Залучення Команди:** Процедури управління сприяють активному залученню команди до процесів прийняття рішень. Розроблення стратегій та планів з урахуванням різноманітних поглядів сприяє вибору ефективних шляхів досягнення мети.

**Ітераційний Підхід:** Застосування ітераційних методів дозволяє постійно адаптувати стратегії та плани відповідно до змінюючихся умов. Це сприяє плавній коригуванню шляху до мети та підтримує стале вдосконалення процесів.

**Використання Технологій:** Використання сучасних інструментів та технологій для збору, аналізу та візуалізації даних полегшує управління невизначеністю та сприяє прийняттю обґрунтованих рішень.

Стратегія Заходів у Невизначеності: Розроблення чітких стратегій та планів дій для реагування на невизначеність. Включення резервних планів та альтернативних шляхів виконання дозволяє уникнути чи зменшити вплив негативних сценаріїв.

Усі ці аспекти свідчать про те, що процедури управління проектами в умовах невизначеності не лише забезпечують структурований підхід до управління, але й надають компаніям можливість ефективно реагувати на виклики, що виникають у ході реалізації проектів.

### **1.3 Постановка завдань менеджменту управління проектами за умов невизначеності даних на основі модифікованих процедур асоціативної пам'яті та редукції математичних моделей класифікації**

В роботі для можливості застосування мережі Хеммінга сформовано кілька представлень та перетворень недетермінованих та слабо структурованих даних, даних що не є точними, до форми процедур мережі Хеммінга. Для кожного з цих представлень проведено широке коло експериментальних чисельних досліджень процесів збіжності та отриманих результатів класифікації. На основі подальшого аналізу результатів експериментів було визначено кращу модель і сформовано нову модифікацію моделі мережі Хеммінга. Необхідно зазначити, що для запропонованої моделі забезпечуються всі властивості щодо реалізації асоціативної пам'яті. Тобто можливість класифікувати неповні або збурені вхідні дані, відповідно до моделей шаблонів, збережених в пам'яті нейронної мережі.

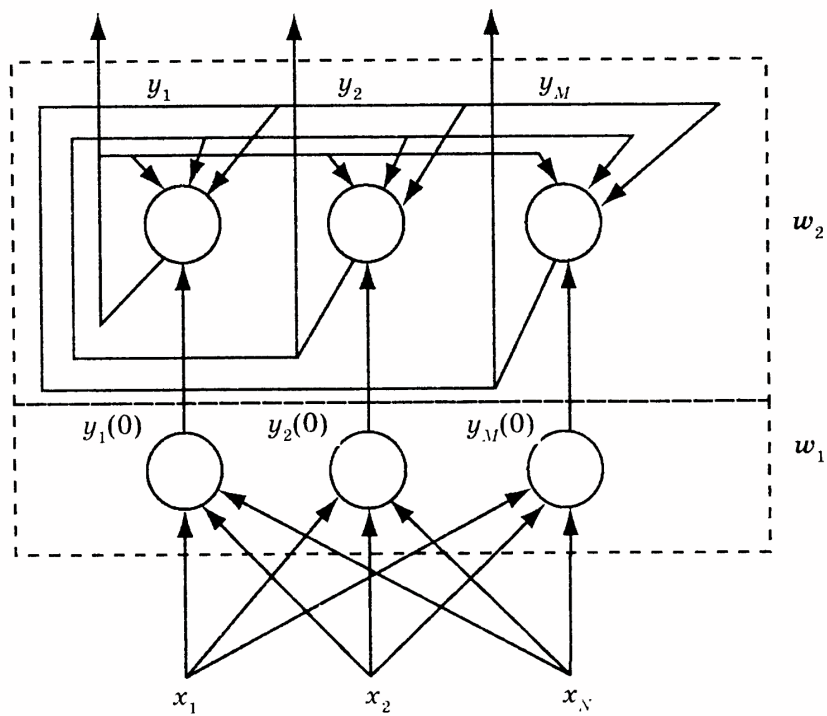


Рисунок 1.1 – мережа Хеммінга

Розглянемо моделі з зворотними зв'язками та процедури нейронної мережі Хеммінга (НМХ) [61, 63], представленої на рис. 1.2 та рис. 1.3. Нейронна мережа НМХ запам'ятовує «к» шаблонів – «п» вимірних векторів  $\{x_i (i=0, 1, n-1) \text{ к-ого зразка}\}$ , які знаходяться в  $\{-1, +1\}$ .

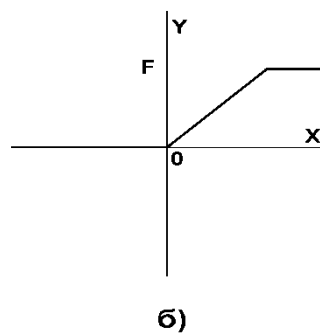


Рисунок 1.2 – Активаційна функція мережі Хеммінга

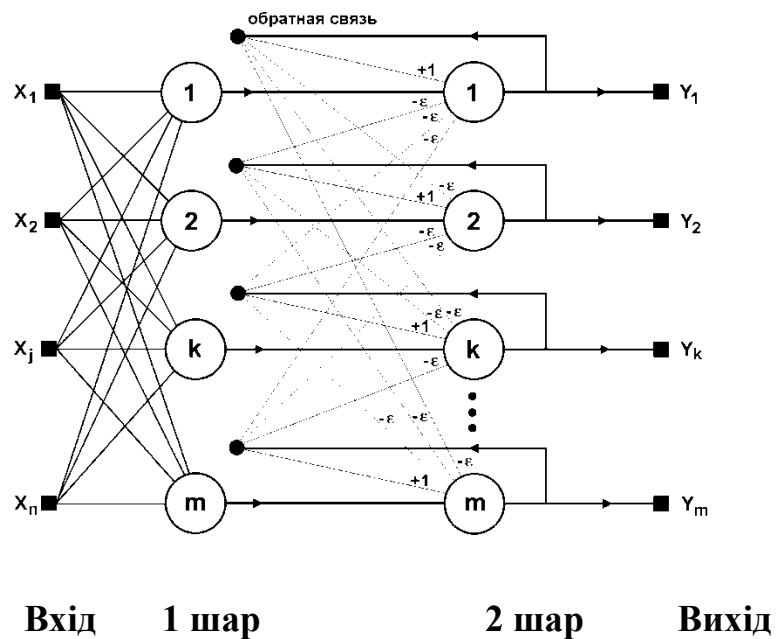


Рисунок 1.3 – Структурна схема мережі Хеммінга

На стадії ініціалізації ваговим коефіцієнтам першого шару і порогу активаційної функції привласнюються наступні значення:

$$w_{ik} = \frac{x_i^k}{2}, \quad i=0\dots n-1, \quad k=0\dots m-1 \quad (1.1)$$

$$T_k = n / 2, \quad k = 0\dots m-1 \quad (1.2)$$

де  $x_{ik}$  –  $i$ -ий елемент  $k$ -ого зразка.

Вагові коефіцієнти гальмуючих (зворотних) синапсів у другому шарі беруть рівними деякій величині  $0 < \varepsilon < 1/m$ . Синапс нейрона, який зв'язаний з його ж аксоном має вагу  $+1$ .

Алгоритм функціонування мережі Хеммінга наступний:

1. На входи мережі подається невідомий вектор  $X = \{x_i; i=0\dots n-1\}$  з елементами множини  $\{-1, +1\}$ , виходячи з якого розраховуються стани нейронів першого шару (верхній індекс у дужках вказує номер шару мережі):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, \quad j=0\dots m-1 \quad (1.3)$$

Після цього отриманими значеннями ініціалізують значення аксонів другого шару:

$$y_j^{(2)} = y_j^{(1)}, j = 0 \dots m-1$$

2. Обчислити нові стани нейронів другого шару:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), k \neq j, j = 0 \dots m-1 \quad (1.4)$$

і значення їх аксонів:

$$y_j^{(2)}(p+1) = f[s_j^{(2)}(p+1)], j = 0 \dots m-1 \quad (1.5)$$

Активаційна функція  $f$  [\*] має вигляд порога (рис. 2 б), причому величина  $Y=F$  повинна бути досить великою, щоб будь-які можливі значення аргументу не призводили до «насичення».

3. Перевірити, чи змінилися виходи нейронів другого шару за останню ітерацію. Якщо так - перейди до кроку 2. Інакше - кінець.

В НМХ порівнюється вихід кожного нейрону (к-ого зразка) з (майже) середнім значенням всіх інших. За рахунок цього визначається найближчий шаблон від вхідного вектору. Збіжність та стійкість цього послідовного процесу є доведеною (у вигляді достатніх умов) для мережі Хопфілда [61] за вимог що на практиці в більшості випадків реалізуються. Модель НМХ має такі самі властивості щодо умов отримання рішень.

Зазначимо що при цьому структура кодів вхідного вектору та зразків не враховується на повторних етапах послідовної процедури (1.8) – (1.9). Для коректного застосування процедури НМХ при інших вхідних векторах, ніж  $\{x_i: i=0 \dots n-1\}$  з елементами множини  $\{-1, +1\}$ , необхідно визначити формальні моделі переходу від прийнятих форм відображення невизначених даних до наведених ( $\{x_i: i=0 \dots n-1\}, -1, +1$ ). Представимо такі формальні моделі переходу докладно для та різних типів відображення

недетермінованих та слабко структурованих даних, «неточно визначених» даних.

## **ВИСНОВКИ**

Дослідження та моделювання процедур менеджменту програмної інженерії при управлінні проектами є важливим кроком у забезпеченні успіху проектів. Це дозволяє розробити чіткі та ефективні процеси, визначити ролі та відповідальність, управляти ризиками та забезпечити ефективну комунікацію. Шляхом дослідження та моделювання процедур менеджменту програмної інженерії, організації забезпечується можливість уникнути проблем та збільшити ймовірність успіху виконання проектів програмної інженерії.

## 2 РОЗРОБКА ТА ДОСЛІДЖЕННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА ПРОЦЕДУР МЕНЕДЖМЕНТУ УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНОЇ ІНЖЕНПЕРІЇ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ДАНИХ НА ОСНОВІ МОДИФІКОВАНИХ МЕРЕЖ ХЕММІНГА ТА ПРОЦЕДУР РЕДУКЦІЇ

### 2.1 Представлення вихідних даних для завдання визначення кандидата на місце.

Претендентами на посаду являються «кандидати»  $K_i$ , які представляють опис кількох своїх виконаних робіт за визначеною темою, портфолію  $P_i$ . Вважається що кожний з кандидатів  $K_i$  має набір реалізованих завдань  $Z_{ik}$  портфолію  $P_i$ , які описані без попередніх вимог до структури опису  $Z_{ik}$ . Також параметри та оцінки за  $Z_{ik}$  можуть відрізнятися для різних  $P_i$  як за кількістю, так і за складом ознак та формою представлення оцінок, у тому числі поданих у формі неточних показників ін. Для менеджменту необхідно розробити та реалізувати процедури щодо забезпечення вирішення завдання класифікації  $K_i$  за неповними та неточно визначеними і збуреними даними.

Остаточну структуру і оцінки щодо представлення  $Z_{ik}$  визначає менеджмент на основі даних  $P_i$ . Для цього необхідно розробити модель та вибрати структуру даних, що забезпечують перетворення  $(P_i, Z_{ik})$  в єдину внутрішню форму представлення. При цьому необхідно урахувати вимогу щодо перетворення всіх параметрів  $(P_i, Z_{ik})$  у єдиний тип – до значень коефіцієнтів упевненості  $CF(A)$  із областю значень  $[-1; +1]$ . Закодовані менеджментом за допомогою значень  $CF(A)$  параметри завдань  $Z_{ik}$  представляють «шаблони»  $Z_{ik}$ , на підставі аналізу яких реалізуються процедури вибору за моделями класифікації. Таким чином, даними для розрахунку являються таблиці шаблонів що кожному «кандидату»  $K_i$  (кількість  $L_i$  при « $m$ » параметрах), закодованих через  $CF(A)$ .

Вимоги до кандидатів Кі («еталон» - «Е») також записуються у форматі CF(A). Кількість та послідовність параметри «Е» співпадає з

Для реалізації завдання із формування таблиць шаблонів кандидатів Кі необхідно розробити спеціалізований інтерфейс.

При цьому також необхідно передбачити ведення бази даних, в якій будуть зберігатися та накопичуватися результати реалізації завдань класифікації на основі моделі НМХ.

Процедури формування бази даних (таблиць) для завдань вибору кандидатів на основі портфоліо.

Процедури База даних містить основні таблиці кандидати (атрибути ознак, номер в системі, число представлених проєктів, тощо).

Проєкти (номер кандидата, номер проєкта, (таблиця, вектор оцінок CFi(A) показника «і»)). Кожний проєкт має всі оцінки для показників, які утворені при вводі. Відсутні показники мають значення «0». Оцінки CFi(A) встановлює «менеджер» на основі опису представлених завдань. Також виставляється отримана при класифікації оцінка даних за проєктом.

Передбачені функції управління даними

Перегляд/відображення кандидатів,

їх проєктів,

поповнення/вилучення, проєктів, кандидатів.

Певна статистика.

Можлива функція управління багатьма призначеннями. В цьому разі необхідно вести параметр «номер призначення» який додається до всіх попередніх таблиць даних. У цьому разі треба корегувати функції управління даними.

Процедура формування складу шаблонів (набір показників).

Попередня таблиця показників «порожня або нульова». Зберігає назви показників для окремого призначення (за номерами). Назви вводить

менеджер на основі опису представлених завдань, назвам присвоюють номери. При формуванні шаблонів використовують таблиці показників для призначення.

## **2.2 Структури шаблонів завдань класифікації.**

Розробляються дві форми представлення шаблонів кандидатів  $K_i$ , а також відповідні ним процедури формування шаблонів класифікації. Шаблони представляють різні форми та моделі описання властивостей «кандидатів»  $K_i$  і вимог до моделі виконавця, тобто «еталону».

В першій процедурі моделі шаблонів представляють певні узагальнені оцінки складових  $Z_{ik}$  набору  $P_i$  (наприклад, середні значення, максимальні чи мінімальні, або встановлені/задані експертом/менеджером).

В другій процедурі форми в якості шаблонів використовуються безпосередньо закодовані структури окремих  $Z_{ik}$ .

В роботі передбачається виконання порівняльного аналізу щодо ефективності наведених процедур та форм утворення шаблонів завдань класифікації на основі модифікованої моделі Хеммінга.

## **2.3 Схема процедури формування шаблонів класифікації зі встановленим рівнем достовірності результатів (Пкап).**

Процедура Пкап (пункт 4) реалізує вимоги математичних моделей Методу Граничних Спрощень (МСП), а також статистики «каппа Коена», (Ккап). На основі МСП (процедурою Пмсп) встановлюється гранична кількість  $n_0$  параметрів моделі регресії (іншої моделі даних досліджень), перевищення якої приводить до формування таких математичних моделей, що не забезпечують результат із встановлено ймовірністю (не відповідність вимог надійності/достовірності результатів та кількості вихідних даних та числа параметрів моделей). Через такі вимоги необхідно вибирати не всі, а найбільше суттєві характеристики процесів, причому не більше  $n_0$ .

В процедурах Пкап також реалізується статистичний критерій, за допомогою визначається рівень «рівнозначності» певних процедур, в тому числі порівнянь кількох експертних тверджень. Статистики «каппа Коена» в процедурах Пкап використовується таким чином: визначається ступінь подібності результатів класифікації на основі моделей шаблонів з різним числом параметрів. В разі подібності результатів вибору із зазначеним ступенем достовірності, визначених за різними моделями-шаблонами, число змінних в моделі первинного набору даних може бути скорочене. При тому необхідно забезпечити (шляхом перебору) вилучення найменше значимих параметрів.

Процедура Пкап використовує нечітку модифікацію моделі мережі Хеммінга НМХ для реалізації завдання класифікації шаблонів за неповними, неточними та збуреними даними.

В якості вихідних даних до процедур Пмсп, Пкап і НМХ надходять закодовані у формі коефіцієнтів упевненості  $CF(A)$  вектори ознак, які представляють шаблони, що визначають як вимоги до кандидатів («еталон» вимог), так і властивості кандидатів, які відповідали (були встановлені) виконанню ними певних проєктів, представлених у портфоліо  $P_i$ . Число параметрів у шаблонах позначене як « $m$ ». Кожному «кандидату»  $K_i$  відповідають кілька ( $L_i$  екземплярів) закодованих шаблонів, також серед них відзначені найкращі (один або кілька) для поточного завдання класифікації («еталон» вимог), яке вирішується. Загальне число шаблонів дорівнює

$$L = \text{SUM} (L_i).$$

Також необхідно задати вимоги щодо достовірності результатів моделювання (позначено « $E$ »).

Далі на основі ( $L$ , « $m$ », « $E$ ») за процедурою Пмсп розраховують значення граничного числа параметрів  $n_0$ .

Процедура формування шаблонів класифікації з встановленим рівнем достовірності результатів « $E$ » призначена для відбору з « $m$ », параметрів

актуальних до кількості  $n_0$  ( $\leq m$ ). Зазначимо, що в якості даних для розрахунку являються таблиці шаблонів при « $m$ » параметрах, що кожному «кандидату»  $K_i$  зберігають  $L_i$  екземплярів закодованих через  $CF(A)$  шаблонів, також для них відзначений найкращий (один або кілька).

Цикл скорочення розмірності шаблонів виконується таким чином. Нехай число параметрів шаблону на етапі  $(t)$  алгоритму дорівнює  $N(t) > n_0$ . При тому  $N(t=0) = m$ , також відомі кращі шаблони з  $L_i$  екземплярів.

Для кожної змінної із числа  $N(t)$  розраховується наступне. Із набору змінних  $N(t)$  та з «еталону» вимог вилучається параметр  $X_j$  (новий вектор шаблону  $(X/ X_j)$ ); виконується класифікація і визначаються найкращі (один або кілька) для нових шаблонів  $(X/ X_j)$ . Для кожного  $K_i$  перевіряються співпадіння нових результатів класифікації для змінних шаблонів при  $(X/ X_j)$  з попередніми, визначеними для всіх « $m$ » параметрів. При співпадінні результатів класифікації за кандидатом  $K_i(P_i)$  відповідна порівняльна оцінка множин наборів  $X$  та  $(X/ X_j)$  отримує значення «+», інакше «-». Результати порівняльного аналізу для всіх змінних утворюють таблицю, наприклад, наступного вигляду

Таблиця 2.1

## Приклад структури моделі класифікації

$L_i / X$	$(X/ X_1)$	$(X/ X_2)$	$(X/ X_3)$	$(X/ X_j)$	...	$(X/ X_{m-1})$	$(X/ X_m)$
5 1	+	-	+	+		-	-
3 2	+	-	+	-		-	-
1 3	-	+	-	+		-	+
2 4	-	+	+	-		+	+
1 5	+	+	-	+		+	-

4	6	+	+	-	-		+	+
---	---	---	---	---	---	--	---	---

В таблиці стовпці містять знак «+» при співпадінні номерів результатів класифікації для наборів параметрів ( $X_i / X_j$ ) з класами першого стовпця ( $X_i$ ), а знак «-» в протилежному випадку.

За результатами табл. 1 за допомогою процедури Пкап, яка реалізує статистику «каппа Коена», для всіх пар змінних ((1, 2), (1, 3), (1, m), (2, 3), ..., (2, m), (3, 4), (3, 5), ..., (3, m), ....., (m-1, m) виконується розрахунок показника Каппа.

Традиційна формула визначення коефіцієнта каппа є такою:

$$K = \frac{P_0 - P_e}{1 - P_e} \quad (2.1)$$

При виконанні циклу розрахунків фіксується пара змінних, для якої Каппа найбільша, а також оцінка (в якісному або текстовому/номінальному вигляді) щодо подібності результатів контрольних розрахунків.

При цьому формується таблиця розбіжностей між первинними та новими результатами класифікації для кожного набору ( $X_i / X_j$ ) за всіма ( $K_i$ ,  $L_i$ ). Наприклад, для  $X_j$  має структуру, за якою розраховується оцінка Каппа,

Таблиця 2.2

Представлення таблиці розбіжностей

$X_j/X_i$	Так	Ні
Так	P1	P3
Ні	P2	P4

де P1, P2 і P3, P4 відповідають рішенням для таблиці 2.1

Якщо рівень показника Каппа відповідає вимогам, одна з змінних цієї пари є зайвою, її властивості представлені у іншій. Тому вона може бути вилучена із списку параметрів «шаблонів» та «еталону», відбувається скорочення розмірності параметрів моделі класифікації. Для вибору змінної

для вилучення (позначимо її  $X^*$ ) можливо використовувати – опитування менеджмента, оцінку кількості вірних результатів (знаків «+» у стовпцях) тощо.

Після скорочення числа параметрів моделі необхідно повторити увесь цикл розрахунків, розпочинаючи з визначення найкращих у  $K_i$  та  $P_i/X^*$ , де –  $X^*$  це вилучений параметр.

Можливий також варіант процедури скорочення (вилучення на кроці кількох змінних), який застосовується при суттєвому перебільшенні числа « $m$ » в умові  $n_0$  ( $\leq m$ ). При цьому для відібраної для вилучення змінної  $X^*$  визначаються пари, для яких (пар  $(X^*, X_i)$ ) показник Каппа( $X^*, X_i$ ) вище встановленого рівня. Така форма функції скорочення параметрів зменшить число кроків процедури спрощення моделі до вимог щодо  $n_0$ .

Схема процедури формування шаблонів діагностування зі встановленим рівнем достовірності результатів (Пкап).

Процедура Пкап використовує нечітку модифікацію моделі мережі Хеммінга НМХ [1] для реалізації завдання класифікації шаблонів за неповними, неточними та збуреними даними. Вона реалізує вимоги математичних моделей Методу Граничних Спрощень (МСП) [3], а також статистики «каппа Коена», Скап [4, 5].

На основі МСП встановлюється гранична кількість  $n_0$  параметрів моделі регресії (моделі класифікації даних ін.), перевищення якої приводить до формування таких математичних моделей, що не забезпечують результат із встановлено ймовірністю (не відповідність вимог надійності/достовірності результатів та кількості вихідних даних і числа параметрів моделей).

Через такі вимоги необхідно вибирати не всі, а найбільше суттєві характеристики процесів, причому не більше  $n_0$ . Для виконання такого вибору пропонується використовувати статистику Скап, щоби визначити рівень «рівнозначності» результатів певних процедур.

При цьому виконується оцінка ступеню подібності результатів класифікації на основі моделей шаблонів з різним числом параметрів. В разі подібності результатів вибору із встановленим ступенем достовірності, визначених за різними моделями-шаблонами, число змінних в моделі може бути скорочене.

При тому необхідно забезпечити (шляхом перебору) вилучення найменше значимих параметрів. В якості вихідних даних до процедур Пкап і НМХ надходять закодовані у формі коефіцієнтів упевненості  $CF(A)$  [6] вектори ознак, які представляють властивості «шаблонів», варіантів класифікації, а також вектор стану, «Е-еталон».

Цикл скорочення розмірності «шаблонів» на основі Скап виконується таким чином. Нехай число параметрів шаблону на етапі (t) алгоритму дорівнює  $N(t) > n_0$ . При тому  $N(t=0) = \langle m \rangle$ , а також відомі кращі рішення для шаблонів розмірності  $\langle m \rangle$ .

Для кожної змінної із числа  $N(t)$  розраховується наступне. Із набору змінних шаблонів  $N(t)$  та з «еталону» вимог вилучається параметр  $X_j$  (новий вектор шаблону  $(X/X_j)$ ); виконується класифікація і визначаються найкращі рішення (один або кілька) для нових шаблонів  $(X/X_j)$ .

Перевіряються співпадіння нових результатів класифікації для змінних шаблонів при  $(X/X_j)$  з попередніми, визначеними для всіх  $\langle m \rangle$  параметрів. При співпадінні результатів класифікації відповідна порівняльна оцінка множин наборів  $X$  та  $(X/X_j)$  отримує значення «+», інакше «-». Результати порівняльного аналізу для всіх змінних утворюють таблицю, в якій стовпці містять дані щодо співпадіння для всіх параметрів моделі (знак «+» при співпадінні результатів класифікації для наборів параметрів  $(X/X_j)$  з класами першого стовпця  $(X)$ , знак «-» в протилежному випадку).

За даними порівняльного аналізу формується таблиця розбіжностей та статистика «каппа Коена» [4, 5]: для всіх пар змінних  $((1, 2), (1, 3), \dots, (1, m), (2, 3), \dots, (m-1, m))$  виконується розрахунок показника Каппа:

При виконанні циклу порівняльних розрахунків фіксується пара змінних, для якої «К» найбільший, а також його оцінка (в якісному або текстовому/номінальному вигляді) щодо подібності результатів контрольних розрахунків.

Якщо рівень показника Каппа (3) фіксованої пари (нехай це такі змінні  $(X/X_j)$  та  $(X/X_k)$ ) відповідає вимогам, одна з змінних цієї пари є зайвою, її властивості представлені у іншій. Прийmemo що це змінна  $(X/X_j)$ . Тому вона «Xj» може бути видалена із списку параметрів «шаблонів» та «еталону», відбувається скорочення розмірності параметрів моделі класифікації.

Можливі два варіанти видалення змінних з пари  $((X/X_j)$  та  $(X/X_k)$ ). Вони відрізняються значеннями, які треба використовувати для наступних кроків циклу в якості еталону при порівняннях.

V1. Якщо видалена змінна це змінна  $(X/X_j)$ , тоді в якості еталону для порівнянь треба використовувати значення саме з стовпця  $(X/X_j)$ . Виконуємо в першу чергу.

V2. Після видалення параметру, наприклад «Xj», результати класифікації для наборів параметрів  $(X/X_k)$ , змінної із пари  $((X/X_j)$  та  $(X/X_k)$ ) що залишився у списку, тобто  $X_k$ , стають новою множиною значень, з якою необхідно виконувати порівняння наборів класифікації  $((X/X_j)/X_p)$  на наступному кроці процедури. Тут  $X_p$  це параметри, для котрих перевіряються класифікації після видалення «Xj». Далі треба буде також перевірити варіант 2 і порівняти результати з V1.

Після скорочення числа параметрів моделі  $(X/X_j)$  необхідно повторити увесь цикл розрахунків до виконання умови ( $m \leq n_0$ ). Можливі варіанти процедури спрощення моделей, які дають на кроці циклу формування вилучення з шаблонів класифікації кількох змінних.

## 2.4 Моделювання та Класифікація в Умовах Невизначеності для Оптимізації Технологічних Процесів

Завдання та процедури класифікації, кластеризації та діагностування за умов неповної визначеності вихідних даних (збурені, неповні ін.) являються досить поширеними на практиці [1]. За їх результатами формуються моделі оптимального керування різноманітними технологічними процесами, регулювання у системах обслуговування, вибору раціональних заходів/виконавців тощо [1]. Через велику складність виробничих, економічних та інших об'єктів керування чи контролю (неповноту даних або обмеженість часу і необхідних для реалізації вибору ресурсів тощо) доступними для аналізу є дані, які необхідно вважати, слабо структурованими, неповністю визначеними, а також лише певними експертними оцінками. Для отримання та реалізації рішень на основі даних з подібними властивостями використовують інтелектуальні моделі і методи. Створення інтелектуальних процедур та інформаційної технології, призначеної для завдань з оптимізації потоків замовлень у обслуговуючих, сервісних C&C, системах запропоновано у [1]. В ній вибір керувань виконується шляхом реалізації завдань діагностування з урахуванням умов невизначеності на основі модифікованої мережі Хеммінга (МХН).

Мета розробки – у реалізації завдань відтворення достовірних багатопараметричних моделей класифікації (ДБМК) при невизначених даних на основі застосування процедур редукції (метод граничних спрощень) та каппа статистики. При цьому математичними моделями ДБМК являються структури параметрів (кількість і склад), тип даних і множина шаблонів, які складають базу даних/знань досліджуваної предметної області.

Удосконалені моделі МНХ мають суттєву відмінність від класичних моделей асоціативної пам'яті Хеммінга в завданнях класифікації

за неповними та збуреними даними. В МНХ використовуються в якості моделей даних нечіткі множини ( $\mu_X: X \rightarrow [0; 1]$ ), а також коефіцієнти впевненості  $CF(A)$  із значеннями в множині  $[-1; +1]$ . При тому класичні моделі Хеммінга представляють дані за допомогою дискретної множини  $\{-1; +1\}$ . Разом з тим у пропонованих нами моделях МНХ залишилось не реалізованим завдання із обґрунтування структури шаблонів багатопараметричної класифікації. Відповідно до нього встановлюється гранична кількість « $n_0$ » параметрів шаблонів, перевищення якої приводить до математичних моделей класифікації, що не забезпечують достовірності результатів діагностування, отримання класифікації із встановленою ймовірністю [3].

Для визначення граничної кількості та складу параметрів моделі класифікації « $n_0$ » в роботі використовуються результати фундаментальної теореми Вапника-Червоненкіса, а також процедури теорії редукції математичних моделей, які визначають метод граничних спрощень (МСП) [3, 10]. Величина « $n_0$ » граничну визначає розмірність простору який синтезується, перевищення якої приводить до втрати гарантій забезпечення заданих вимог достовірності результатів класифікації (ймовірність помилкової класифікації не перевищує задане значення). З методом МСП, а також його формою у вигляді  $\alpha$ -процедури, будується простір встановленої розмірності, в якому подібні («схожі») об'єкти розділяються лінійним правилом рішення. За рахунок чого визначається та відновлюється закономірність схожості. При цьому використовуються показники «розділяючої сили» признаков, які утворюють простір моделі індикаторної функції.

У завданнях пошуку закономірностей розглядають такі: рівності (встановлюють функціональну залежність), схожості (класифікація, віднесення об'єктів до певного класу схожих чи подібні між собою), закономірності порядку (послідовності об'єктів відповідно встановленої

властивості). При визначенні закономірностей схожості дані спостережень призначені для відновлення певної індикаторної функції

$$y=F(X'), \quad (2.2)$$

де  $y$  – цільовий параметр, а  $X=(x_1, x_2, \dots, x_m)$  – вектор, значення якого впливають на цільовий параметр, функція  $F$  відповідає типам розшукуваних закономірностей.

Принцип редукції (ПР) визначає, що при аналізі емпіричних даних до моделей (1) обираються такі аргументи (властивості) « $x_i$ », за допомогою яких формуються найбільш прості закономірності. За ПР замість побудови складних «придуманих» функцій у випадковому просторі розмірності « $m$ », до моделі (1) обираються лише такі  $n_0 \ll m!$  параметрів « $x_i$ », в просторі яких «проста» залежність  $F(X)$  є найбільш змістовною. Тобто, первинним і більш складним являється завдання вірного визначення координатного простору. Визначення форми закономірності « $F$ » є вже наступним та більш вивченим етапом. В завданнях класифікації моделей (ДБМК) при невизначених даних, що досліджуються у статті, в якості моделей предметної області використовуються набори шаблонів/еталонів ознак, які визначають певні класи схожих чи подібні між собою об'єктів (станів, умов, керувань тощо), відносно них і проводиться аналіз вхідних наборів.

Теорема Вапника-Червоненкіса встановлює наступне твердження

$$\varepsilon \leq (\ln(N) - \ln(h))/L, \quad (2.3)$$

за яким встановлюється оцінка ймовірності  $(1 - h)$  щодо умови безпомилкового розділення випадкової і незалежної вибірки довжини « $L$ » при заданій граничній величині помилкової класифікації « $\varepsilon$ ». У (2) параметр « $N$ » це число способів, якими будь яку вибірку довжини « $L$ » можна розділити на дві підмножини даним класом правил вирішення (формул, моделей). Для заданих значеннях « $\varepsilon$ » та « $h$ » можливо розрахувати

мінімально допустиму довжину вибірки, необхідну для формування (1).

Величина

$$n_0 = (\varepsilon * L + \ln(h)) / \ln(m) \quad (2.4)$$

визначає розмірність синтезованого простору, перевищення якої призводить до втрати гарантії досягнення заданих параметрів « $\varepsilon$ ,  $h$ » щодо достовірності (1).

В разі невідповідності структури шаблонів багатопараметричної класифікації ДБМК вимогам достовірності (відновлення закономірності схожості) необхідно вибирати не всі, а найбільше суттєві характеристики процесів, причому не більше « $n_0$ ». В цій статті розроблена нова процедура, яка забезпечує формування простору багатопараметричної класифікації з указаними вимогам достовірності. Змістовно ця процедура враховує, що при виборі параметрів шаблонів для завдань класифікації необхідно враховувати можливість існування залежності між змінними. Узагальнено такі відносини можливо розглядати як певну «подібність або схожість, заміну» параметрів щодо результату. Для оцінки таких властивостей параметрів моделей класифікації у статті використовується статистичний критерій каппа Коена [4, 5]. За каппа статистикою визначається оцінка рівню «рівнозначності» результатів, отриманих шляхом виконання різних процедур. В якості таких порівнюваних результатів можуть бути також твердження експертів [5]. При застосуванні ПР розмірності каппа статистикою розраховується оцінка результатів класифікації за моделями шаблонів з різним числом чи складом параметрів. В разі «подібності» результатів вибору для конкуруючих шаблонів при заданих параметрах « $\varepsilon$ ,  $h$ » число змінних в моделі класифікації можна скоротити. Також необхідно забезпечити формування шаблонів з найменше подібних параметрів предметної області аналізу.

Представимо схематично процедуру формування шаблонів ДБМК класифікації зі встановленим рівнем достовірності результатів (ПРК). Етапи ПРК наступні:

- 1) виконати оцінку показника «каппа» ступеню подібності результатів класифікації на основі моделей шаблонів з різним числом або складом параметрів,
- 2) при забезпеченні зі встановленим ступенем достовірності подібності результатів класифікації для різних моделей-шаблонів можливо виконати скорочення моделі, залишити один із шаблонів,
- 3) визначити і видалити найменше значимі або найбільше «подібні» між собою параметри моделі класифікації, враховуючи значення « $n_0$ ».

Приведемо узагальнений варіант циклу скорочення розмірності «шаблонів» моделі при заданому входному «еталон»/»вимоги». Позначимо як  $N(t) > n_0$  число параметрів шаблону на етапі (t), а їх розмірність при  $N(t=0) = «m»$ . Утворюється множина конкуруючих моделей шаблонів меншої розмірності. Для цього з набору змінних шаблонів  $N(t)$  і «еталону» вимог вилучається, наприклад, параметр  $X_j$ ; нові вектори для цих спрощених моделей шаблонів позначимо  $(X/X_j)$ . Такі спрощені моделі формуються кожної змінної із числа  $N(t)$ , а за ними виконується класифікація на основі МНХ.

Перевіряються співпадіння («подібність» за каппа статистикою) нових результатів класифікації для усіх пар змінних шаблонів, утворених на основі  $(X/X_j)$ . Для розрахунку показника подібності «каппа» результатам класифікації за різними моделями шаблонів приписується порівняльна оцінка. Для різних наборів змінних  $(X/X_j)$  шаблонів виконується порівняння, і для кожного шаблону призначаються значення «+» або «-» в залежності від результатів класифікації. На основі порівняльного аналізу для всіх змінних утворюють таблицю розбіжностей, за якою розраховують статистичні оцінки «каппа Коена» [4, 5]:

$$K = (P_0 - P_e) / (1 - P_e), \quad (2.5)$$

В рівнянні  $P_0$  – ймовірнісна оцінка що показує наскільки спостережувана узгодженість краща за випадкову, а  $P_e$  – представляє результат підрахунку максимально можливої узгодженості за винятком випадкової узгодженості конкуруючих тверджень (тут моделей шаблонів) [4, 5]. Відзначимо, що формування таблиць розбіжностей можливе за різними схемами завдань класифікації за МНХ.

Для вибору параметра моделі  $N(t)$  в циклі розрахунків визначається пара наборів змінних виду  $(X/X_j)$  із найбільшим значенням « $K$ » (1), а також відповідна експертна оцінка «подібності» скорочених наборів  $(X/X_j)$  (в текстовому/номінальному вигляді). Якщо величина оцінки пари (1) відповідає вимогам, вважається що властивості одного набору даних достатньо представлені в іншому. Тому певна змінна/їх множина може бути видалена з «шаблонів» та «еталону», відбувається скорочення розмірності параметрів моделі. Після видалення визначених параметрів з моделі класифікації увесь цикл розрахунків повторюється до виконання умови (« $m$ » $\leq n_0$ ).

## ВИСНОВКИ

У цьому розділі було проведено моделювання та експериментальний аналіз процедур мережі Хеммінга для класифікації недетермінованих та слабо структурованих даних.

Була розглянута проблема представлення вихідних даних для завдання визначення кандидата на місце. Визначено важливі характеристики та параметри даних, які впливають на результат класифікації.

Описано структури шаблонів, використаних для класифікації. Визначено ключові властивості, які враховуються при визначенні шаблонів для ефективної класифікації даних.

Представлено схему процедури формування шаблонів класифікації зі встановленим рівнем достовірності результатів (P<sub>кап</sub>). Розглянуто, які фактори впливають на формування надійних шаблонів класифікації.

Проведено моделювання та класифікація в умовах невизначеності для оптимізації технологічних процесів. Розглянуто методи та алгоритми, що використовуються для ефективної класифікації в умовах невизначеності, зокрема в контексті оптимізації технологічних процесів.

Загалом, результати експериментального аналізу свідчать про ефективність використання процедур мережі Хеммінга для класифікації недетермінованих та слабо структурованих даних в умовах невизначеності. Описані процедури та методи можуть бути корисними у сферах, де важлива точність та надійність класифікації при обробці невизначених даних.

### 3 ФОРМУВАННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ЗАВДАНЬ МЕНЕДЖМЕНТУ УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНОЇ ІНЖЕНПЕРІЇ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ДАНИХ

#### 3.1 Постановка та формалізація завдань дослідження інтелектуальних процедур нейронної мережі Хеммінга з неточно визначеними даними

Коротко вкажимо основні відмінності мережі Хеммінга (МХ), коли дані про зразки (модель) і вхідний сигнал кодують за допомогою дискретної множини  $\{-1; +1\}$  [10 МХ], від алгоритму моделей МХН [1, 8] з моделями даних нечітких множин ( $\mu_X: X \rightarrow [0; 1]$ ).

Мережа МХ складається з двох шарів, які мають по «m» нейронів, де «m» - число зразків, які слід враховувати при виборі класу/управління. Нейрони першого шару мають по «n» синапсів (вхідних векторах), з'єднаних з входами мережі. Нейрони другого шару пов'язані між собою інгібіторними (із знаком мінус, зворотними) синаптичними зв'язками. Вагові інгібіторні коефіцієнти у другому шарі вибирають за умов  $0 < \varepsilon < 1/m$ . Передбачений єдиний синапс (вхід) з позитивним (вага +1) зворотним зв'язком для кожного нейрона з'єднаний з його ж аксоном, виходом. Ідея роботи МХ полягає в розрахунку відстані Хеммінга (число відповідних за позицією бітів в двох бінарних векторах, що відрізняються між собою) від вхідного/тестованого образу до всіх зразків (шаблонів/класів, умов функціонування тощо). МХ вибере зразок з мінімальною відстанню Хеммінга до вхідного сигналу. При тому буде активізований тільки один вихід мережі, що відповідає цьому зразку – мережа визначила клас для вхідного набору.

Пояснимо головну відмінність алгоритм функціонування мережі МХН у порівнянні з МХ. При подачі на входи мережі невідомого вектору X

=  $\{x_i: i=0 \dots n-1\}$  розраховуються стани нейронів першого шару (верхній індекс у дужках):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, \quad j=0 \dots m-1; \quad (3.1)$$

величинами (1) ініціалізують значення аксонів другого шару:  $y_j(2) = y_j(1)$ ,  $j = 0, \dots, m-1$ , за яких виконуються подальші ітерації. У (1) величини ( $x_{ik}$  –  $i$ -ий елемент  $k$ -ого зразка)

$$w_{ik} = \frac{x_i^k}{2}, \quad i=0 \dots n-1, k=0 \dots m-1 \quad (3.2)$$

являються ваговим коефіцієнтам першого шару. З урахуванням дискретної множини можливих значень моделі МХ  $\{-1; +1\}$  сума (1) представляє відстань Хеммінга. При переході до нечітких множин значень  $x_{ik} - (\mu_X: X \rightarrow [0; 1])$  необхідно іншим чином виконувати розрахунок відстані від вхідного образу до всіх зразків. Також треба забезпечити значення відстаней відповідно (1). Для оцінки величин відстані між нечіткими елементами ( $\mu_X$ ) зразків і вхідним вектором  $X = \{x_i: i=0 \dots n-1\}$  було уведено нечітке відношення наступного виду

$$R(W, X) = \{\mu_R(w_i, x_i) / \{w_i, x_i\} = (1 - \text{abs}(w_i - x_i)) / \{w_i, x_i\}, i=0, 1, \dots, n-1. \quad (3.3)$$

Якщо значення ступенів приналежності величин  $\{w_i, x_i\}$  однакові, тоді  $\mu_R(w_i, x_i) = 1$ . Коли одина величин з  $\{w_i, x_i\}$  дорівнює 0, а інша 1  $\mu_R(w_i, x_i) = 0$ , інакше  $\mu_R(w_i, x_i) \rightarrow [0; 1]$ .

Відношення (3) представляє одну із можливостей реалізації МХН для формування нечітких моделей класифікації на основі асоціативної пам'яті при збереженні структури та можливостей алгоритмів МХ. Подібним чином формуються структури рівнянь виду (1) для завдань класифікації моделей об'єктів, представлених коефіцієнтами впевненості  $CF(A)$ . Числові дослідження [1, 8] показали коректність та числову ефективність МХН.

Представлено моделі МХН класифікації як нечітких, так і завдань з моделями на основі коефіцієнтами впевненості  $CF(A)$  (завдання щодо призначення на посаду на основі резюме та портфоліо).

Розглянемо завдання розробки процедури, призначеної для формування ефективних шаблонів класифікації моделей МХН при урахуванні можливостей методу граничних спрощень і каппа статистики. Головними складовими цієї процедури являються питання щодо вибору і представлення структури моделі системою шаблонів/зразків, вибір множини моделей конкуруючих процесів або показників для проведення порівняльних досліджень «подібності» на основі каппа статистики Коена, виконання порівняльного аналізу конкуруючих моделей, визначення структур або параметрів для спрощення (видалення зі складу моделей класифікації МХН).

Приведемо умовний приклад реалізації наведених вище завдань на основі табл. 1 результатів класифікації для шаблонів з параметрів ( $X_1, X_2, \dots, X_6$ ). В табл. 1 позначено :  $K_i$  – класи/заски моделі, які можуть містити кілька екземплярів  $L_i$ ;  $X$  – набір параметрів моделі класифікації на поточному етапі процесу редукції;  $X/\{X_j\}$  – скорочені набори моделей класифікації та вхідних векторів на поточному етапі без множини параметрів  $\{X_j\}$ , які видалені з моделей; знаки «+» – визначення модифікованих шаблонів переможців (класів) при застосуванні моделі МХН. У стовпці  $X$  позначаються шаблони, які були визначені на основі  $X$  наборів параметрів. Саме серед скорочених моделей шаблонів  $X/\{X_j\}$  визначається набір параметрів  $\{X_j\}$ , які необхідно видалити на наступному етапі процедури редукції. В загальному випадку множини параметрів  $\{X_j\}$  можуть містити не одну, а любую кількість змінних, встановлену у завданні. Прийняте в табл. 1 число параметрів моделі класифікації  $n=6$  – умовне, а граничне значення  $n_0 = 4$ . Дані табл. 1 фіксують результати класифікацій за мережею МХН, які отримані при однакових вихідних постановках завдань для всіх конкуруючих варіантів, сутність яких зараз не суттєва, не розкривається. Метою прикладу є демонстрація фрагментів процедури редукції для завдань класифікації з використанням статистики каппа Коена, яка використовується подібно до показників «розділяючої сили» у  $\alpha$ -процедурі [3, 9].

### 3.2 Аналіз Конкуруючих Варіантів та Методів Оцінки Моделей Класифікації

Приведемо приклади завдань класифікації/діагностування за недетермінованими даними, які можуть бути реалізовані засобами МХН з використанням алгоритмів редукції простору моделі та каппа статистики при нечітких, а також даних у форматі коефіцієнтів упевненості  $CF(A)$ .

Оцінки розмірності простору математичних моделей завдань класифікації відповідно до вимог процедур редукції (2), (3) приведені у табл.

3.1

Таблиця 3.1

Розрахунки граничної розмірності простору моделей класифікації завдань ЗАТ

вибірка	помилка	достовіру	парам	розмірн	помилка	n0
L	eps	(1-h)0/h	m	n0	eps	
30	0,1	0,1	5	0,43		
40	0,2	0,2	10	2,80		
40	0,2	0,2	6	3,56		
300	0,1	0,1	10	12,03		
300	0,2	0,1	10	25,06		
500	0,2	0,1	10	42,43		
500	0,1	0,2	10	21,02		

Гранична розмірність  $n_0 = 58$  для вимог останнього рядку потребує  $L = 1200$ .

Модель класифікації при даних у форматі коефіцієнтів упевненості  $CF(A)$  змістовно відповідає завданню менеджменту – відбір/призначення одного кандидата із зазначеної множини (ЗК). Наведене змістовне завдання призначення [12] відоме, має багато моделей і процедур реалізації, в залежності від типу та ознак параметрів, способів їх отримання і оцінювання, а також ступеня достовірності даних про «кандидатів» і вимог вхідного «еталону». Особливість завдання ЗК щодо призначення, яке досліджене у цій роботі, складається із наступного. Вважається, що менеджментом реалізується процедура відбору «кандидата» на основі даних його портфоліо (переліку виконаних і представлених робіт у певних проєктах). При цьому кожний з кандидатів  $K_i$  має набір реалізованих завдань

Зік портфоліо  $P_i$ , які описані без попередніх вимог до структури опису Зік. Також параметри та оцінки за Зік можуть бути представлені текстом/мовна-форма, відрізняються для різних кандидатів  $P_i$  як за кількістю, так і за складом ознак і формою представлення оцінок, у тому числі поданих у формі неточних показників ін. Таким чином, маємо завдання класифікації  $K_i$  за неповними та неточно визначеними і збуреними даними. Остаточну структуру і оцінки щодо представлення Зік визначає менеджмент, який встановлює оцінки всіх ознак у формі коефіцієнтів упевненості  $CF(A)$  на підставі завдань Зік портфоліо  $P_i$ . Таким чином, шаблони/зразки різних кандидатів  $K_i$  у моделі завдання ЗК суттєво неоднорідні, можуть мати «пропуски» в ознаках, число шаблонів кожного кандидата  $K_i$  різне. Вхідний «еталон вимог» містить всі характеристики, ураховані та означені «менеджментом» при формуванні моделі ЗК за даними портфоліо. Метою розрахунку являється визначення шаблону/зразка, який в найбільшому ступені відповідає «еталон вимог», тобто виконувалося подібне завдання..

Разом з тим у завданні ЗК також необхідно передбачити дві процедури формування шаблонів класифікації, тобто моделей описання властивостей «кандидатів»  $K_i$  і вимог до моделі виконавця, заданого параметрами вхідного «еталону». В першій процедурі шаблони представляють певні узагальнені оцінки складових Зік набору  $P_i$ . В другій процедурі в якості шаблонів використовуються безпосередньо закодовані структури окремих Зік. Також треба виконати порівняльний аналіз щодо ефективності наведених процедур та форм утворення шаблонів завдань класифікації на основі модифікованої моделі Хеммінга МХН. При формуванні шаблонів для класифікації «кандидатів»  $K_i$  необхідно забезпечити вимоги щодо встановлених ймовірнісних оцінок достовірності рішення. Для цього застосовуються приведені методи та алгоритми редукції простору класифікації та статистика «каппа» Коена, які забезпечують формування шаблонів моделі класифікації  $K_i$  з числом параметрів, що не перевищує граничну межу  $n_0$ .

Відзначимо, що при представленні даних (системи зразків), а також результатів класифікації у формі табл. 1, виконанні всі формальні вимоги до моделей класифікації та їх елементів для завдань із вибору виконавців ЗК.

Приведемо приклад постановки ЗК (для кандидатів  $K_1, \dots, K_4, \dots, K_{ij}$ ) з неточно визначеними та природньомовними даними, які кодуються у формі коефіцієнтів  $CF(A)$  із значеннями у  $[-1; +1]$  табл. 7. За табл. 7 кандидати мають різну кількість зразків, певні зразки не мають деяких ознак (наприклад,  $K_{11}$  і  $K_{12}$  не мають даних для  $X_5$  і  $X_{11}$ , а  $K_{41}$  не містить  $X_5, X_6, X_9$  та  $X_{11}$ ). У «еталон вимог» включені наступні ознаки: 1) Пріоритет задачі що виконувалась. 2) Складність задачі. 3) Оцінки навичок категорії  $H_1$ . 4) Оцінки навичок категорії  $H_2$ . 5) Оцінка рівню певного фаху 6) Оцінка досвіду виконавця таких завдань. 7) Оцінка рівня визначених знань  $Z_1$ . 8) Завантаженість у проєкті. 9) Навичка  $H_9, \dots, 11$ ) Оцінка навичок контролю. Структура шаблонів моделі для кожного представленого завдання кандидатами ( $K_i$ ) визначається наборами показників  $CF(A)$ , визначеними менеджментом за Зік портфолію  $P_i$ . Засобами моделі МХН [1, 8], а також процедурам редукції та каппа Коена, необхідно встановити шаблон індивідуальних ознак Зік «виконавця, клас  $K_i$ » у форматі  $CF(A)$  для  $(X_1, \dots, X_{11})$ , який найкращим чином відповідає вхідній моделі «еталон вимог» вигляду, наприклад,  $(X_1=0.9, X_2=0.75, X_3=-0.5, \dots, X_6=1, \dots, X_{11}=-0.3)$ . Як було відзначено вище, структура простору моделі класифікації формується окремо для кожного «еталону вимог», щоб забезпечити встановлену достовірність результату класифікації.

Таблиця 3.2

Приклад моделі зразків для завдань ЗК претендентів  $K_1, \dots, K_4, \dots, K_{ij}$

	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$	$X_9$	$X_{10}$	$X_{11}$
$K_{11}$	0.5	0.8	0	0.4	0.7	1	0.8	0.1	0	0.1	0.3
$K_{12}$	0.3	0.9	1	0.5	0.1	1	0.1	0.2	1	0.3	0.7
$K_{13}$	0.7	0.5	0.8	0.3	0	1	0.4	0.7	0.6	0.3	0
$K_{21}$	0.4	0.9	0.4	-1	0.7	0	0.4	0.9	0.4	0	0.8

$K_{22}$	0.4	0.9	0.4	0	0.8	1	0.6	0.5	0.3	0	0.6
$K_{31}$	0.2	0.7	2	0.3	0	0,5	0.2	0.7	1	0.3	1
$K_{32}$	0.5	0.8	3	0.4	0.5	0,7	0.1	0.4	0.5	0.6	0.7
$K_{33}$	0.1	0.2	0	0.8	0	0	0.4	0.6	0	0.5	0
$K_{41}$	0.5	0.8	0	0.4	0.7	1	0.8	0.1	0	0.1	0.3
$K_{ij}$	...	...	...				...		...		...

### 3.3 Програмне Забезпечення та Процедури Редукції Розмірності Моделей для Управління Проектами в Області Програмної Інженерії

Для реалізації завдань призначення на основі  $P_i$  та моделювання процедур менеджменту програмної інженерії при управлінні проектами розроблене відповідне програмне забезпечення.

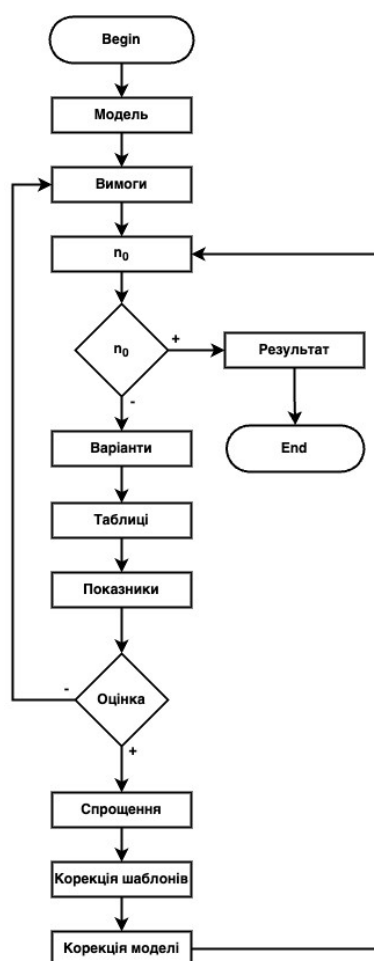


Рисунок 3.1. Блок-схема алгоритму процедури редукції розмірності моделі

За схемою рис. 3.1 узагальнено алгоритм редукції на основі каппа статистики виконується таким чином. Отримується Модель завдання класифікації за умов невизначеності даних, визначаються Вимоги щодо забезпечення точності та достовірності очікуваних результатів класифікації, за отриманими даними розраховується гранична розмірність моделі  $n_0$ , перевіряється відповідність поточної моделі класифікації вимогам до моделі, за значенням  $n_0$ , в разі виконання вимог – маємо достовірний Результат за вимогами, у разі невиконання вимог починається аналіз простору і формування конкуруючих Варіантів спрощених моделей класифікації, для всіх сформованих варіантів реалізації конкуруючих моделей формуються таблиці розбіжностей, за таблицями розраховуються показники статистики каппа Коена та визначається загальна оцінка всіх варіантів, якщо така оцінка схожості конкуруючих моделей не відповідає вимогам достовірності, необхідно змінити вимоги, інакше виконується процедура спрощення (видалення певної структури параметрів моделі класифікації), при тому перевіряється необхідність і виконується корегування систем шаблонів, а також моделі класифікації

#### Специфікація процедури "Каппа Коена"

Мета: Визначення результату класифікації "кандидатів"  $K_i$  з використанням "Каппа Коена" з урахуванням ймовірнісних вимог та методів спрощень (МСП).

Вхідні параметри:

- Кількість кандидатів ( $K_i$ ).
- Дані портфоліо кандидатів ( $Z_{ik}$ ) у формі нечітких даних та коефіцієнтів упевненості  $CF(A)$ .
- Моделі описання властивостей "кандидатів"  $K_i$  (шаблони класифікації).
- Вимоги до моделі виконавця ("еталон").
- Ймовірнісні вимоги для класифікації результату.

Вихідні параметри:

- Результат класифікації "кандидатів" Кі з урахуванням імовірнісних вимог.

Кроки процедури:

1. Формування шаблонів класифікації "кандидатів" Кі з використанням методів спрощень (МСП) та статистики "Каппа Коена".
2. Класифікація кандидатів на основі шаблонів та методу "Каппа Коена".
3. Оцінка ступеня згоди між результатами класифікації та вимогами еталону з використанням "Каппа Коена".
4. Врахування імовірнісних вимог для визначення кінцевого результату класифікації.
5. Виведення результату класифікації з урахуванням імовірнісних вимог.

Специфікація n0 для обмеження кількості параметрів:

Назва параметру: n0 (гранична межа кількості параметрів).

Тип параметру: Ціле число.

Опис параметру: Параметр n0 обмежує максимальну кількість параметрів, які можуть бути використані при формуванні шаблонів класифікації "кандидатів" Кі. Це обмеження визначається менеджментом програмної інженерії та може залежати від специфіки проєктів та потреб бізнесу.

Значення параметру n0:

$n0 = [\text{ціле число}]$  (де це число встановлюється відповідно до вимог менеджменту та бізнес-потреб).

Ця гранична межа n0 буде використовуватися для формування шаблонів класифікації "кандидатів" Кі таким чином, щоб кількість параметрів не перевищувала зазначеного значення n0.

Визначається за формулою  $n0 = (\epsilon * L + \ln(h)) / \ln(m)$

де L – довжина вибірки,  $\epsilon$  – ймовірність помилкової класифікації за правилом, h – достовірність твердження щодо висновку про властивості

результату,  $m$  - число параметрів, первинних властивостей, за якими проведені спостереження для  $L$  –вибірки

Мета програми: Забезпечення обґрунтованого призначення кандидатів на ролі та проекти в галузі програмної інженерії, використовуючи аналіз портфоліо та методи спрощень та статистику "Каппа Коена".

функціональність:

#### 1. Завантаження даних:

Можливість завантажувати дані про "кандидатів"  $K_i$ , включаючи їхнє портфоліо  $P_i$  та оцінки.

Підтримка різних форматів для представлення даних, включаючи нечіткі дані та коефіцієнти упевненості.

#### 2. Формування шаблонів:

Розробка двох процедур формування шаблонів класифікації "кандидатів"  $K_i$ : узагальнені оцінки складових  $Z_{ik}$  набору  $P_i$  та закодовані структури окремих  $Z_{ik}$ .

Забезпечення можливості враховувати різні ознаки та параметри при формуванні шаблонів.

#### 3. Класифікація кандидатів:

Реалізація модифікованих мереж Хеммінга для класифікації на основі шаблонів та методу "Каппа Коена".

Оцінка ступеня згоди між результатами класифікації та вимогами еталону з використанням "Каппа Коена".

#### 4. Візуалізація результатів:

Виведення результату класифікації у зрозумілому вигляді, наприклад, у вигляді рейтингу або списку призначених кандидатів.

інтерфейс:

Графічний інтерфейс користувача (GUI) для завантаження даних та налаштування параметрів.

Можливість визначення кількості параметрів, що не перевищує граничну межу  $n_0$ .

Інтуїтивно зрозумілий інтерфейс для виведення результатів класифікації та їхньої інтерпретації.

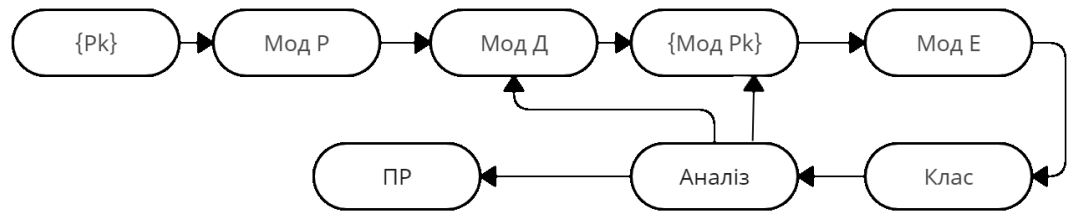


Рис 3.2 Схеми ІТ управління процесами призначення кандидатів

Блоки структури ІТ визначають наступні процедури:

- {Pk} – отримання даних портфоліо/резюме кандидатів;
- МодР – формування загальної моделі даних портфоліо/резюме конкурсу кандидатів;
- МодД – вибір типу моделі представлення даних МодР (нечікі величини/коэф.впевненості);
- {Мод Рк} – формування моделей кожного кандидату у форматі МодД;
- МодЕ – представлення еталону вимог у форматі МодД;
- Клас – реалізація завдання класифікації на моделях {Мод Рк, }, результат P\*k,;
- Аналіз – визначення ступеня однозначності та достовірності результатів класифікації, прицняття релътату / визначення необхідності корегування моделей кандидатів {Мод Рк}/ або корегування загальної моделі даних портфоліо/резюме МодР.

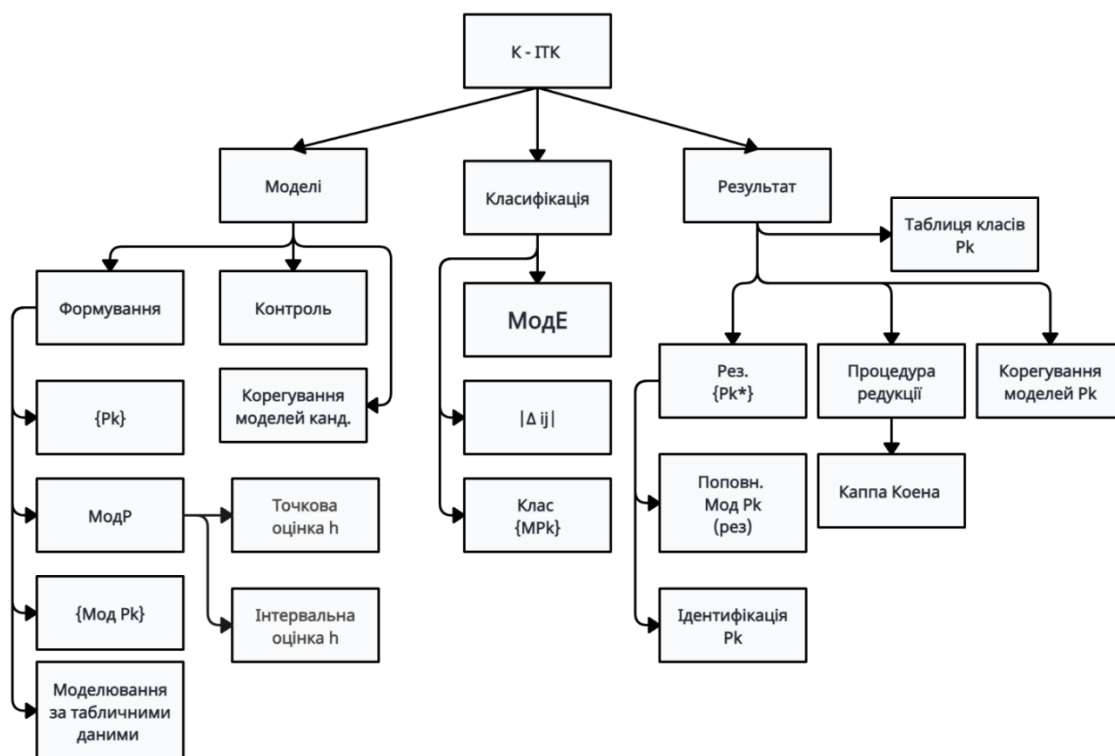


Рис. 3.3 Структура програмного комплексу ІТ управління процесами вибору кандидатів

Б1 Моделі – Формування, контроль, корегування моделей кандидатів {Мод Рk}

Б2 Класифікація – процедури визначення кандидата-переможця на основі модифікованої мережі Хемміга

Б3 Результат – предствлення результатів вибору кандидата-переможця.

Б 1.1 Формування - {Pk}, МодР, {Мод Рk}

Б 1.2 Контроль

Б 1.3 Корегування моделей кандидатів

Б 2.1 МодЕ – представлення еталону вимог у форматі МодД;

Б 2.2 Моделі класифікації

Б 2.3 Клас – реалізація завдання класифікації на моделях {Мод Рk, }, результат Р\*k,;

Б 3.1 Таблиця класів Р k

Б 3.2 Результат Рk \*

### Б 3.3 Корегування Мод Рк

Б 1.1.1 {Рк} – отримання даних портфоліо/резюме кандидатів

Б 1.1.2 МодР – формування загальної моделі даних портфоліо/резюме конкурсу кандидатів

Б 1.1.3 {Мод Рк} – формування моделей кожного кандидату у форматі МодД;

Б 3.2.1 Поповнення Мод Рк та виведення результатів

Модель і процедура отримання експертних оцінок показників моделей кандидатів {Мод Рк}.

В якості вихідних даних до процедур класифікації на основі модифікованих моделей МХН надходять закодовані у формі нечітких величин (НВ) або коефіцієнтів упевненості  $CF(A)$  вектори ознак моделі класифікації. Такі вектори представляють властивості «шаблонів» {Мод Рк}, варіантів класифікації, а також векторів вимог – «Е-еталон». Завдання полягає у визначенні «найближчого» до вектору «Е-еталон» шаблону серед всіх {Мод Рк}.

У роботі приведено дві моделі та процедури отримання експертних оцінок показників моделей кандидатів {Мод Рк}. За ними кожному показнику загальної моделі даних портфоліо/резюме МодР призначається (експертом/менеджером) оцінка показника кандидата {Рк} у формі нечітких величин або коефіцієнтів упевненості  $CF(A)$ .

У першій моделі оцінювання МО1 значення показника вводиться безпосередньо, враховуючи область можливих значень НВ і  $CF(A)$ .

У другій моделі оцінювання МО2 значення показника визначається на основі нечіткої моделі «Показник» (МПок), призначеної для формування значень на основі опитування (експерта/менеджера). В Мпок використовується наступна нечітка апроксимація області значень можливих оцінок НВ. Приведемо модель МПок для визначення нечітких оцінок. Модель апроксимації області значень НВ приведена на рис. 3.4, а правила бази правил для формування значень на основі опитування – на рис. 2.

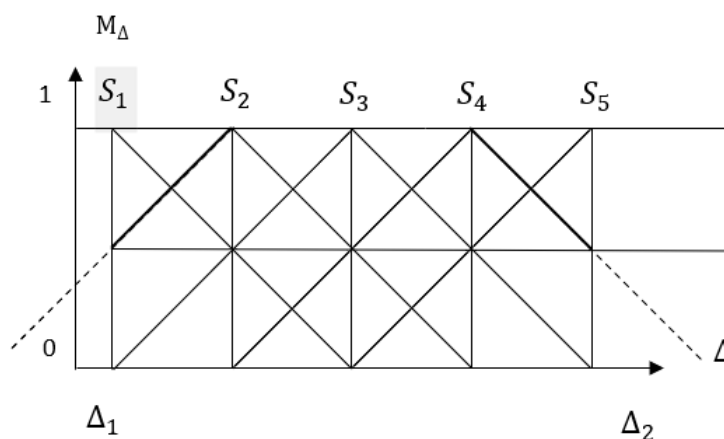


Рисунок 3.4 Варіант системи трикутних моделей для апроксимації області оцінок нечітких показників (інтервалів значень показників)

Тож, область значень показника має апроксимацію трикутними величинами  $S = \{S_1, S_2, \dots, S_5\}$ .

На рис. 2 приведені одні з правил для  $S$ , у лівій частині «Т» показані області можливих значень оцінок показника, а у правій – нечітка модель значення показника. Мета правила – на основі наближеного діапазону показника «Т» розрахувати  $\alpha$ -рівень, за яким визначається модель області значень характеристик «V», тобто оцінок для показників моделей  $\{\text{Мод } P_k\}$ .

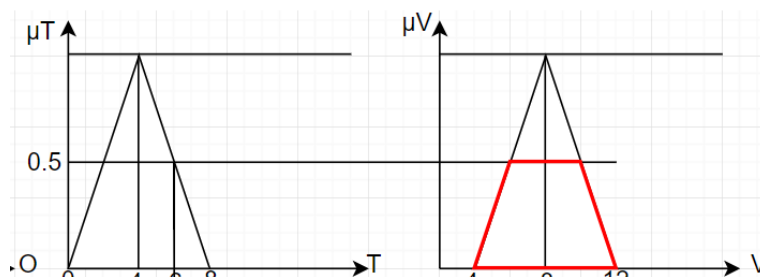


Рисунок 3.5 Структура правил оцінювання ступенів реалізації моделей апроксимації діапазонів НВ  $\{S_1, S_2, \dots, S_5\}$  рис. 1.

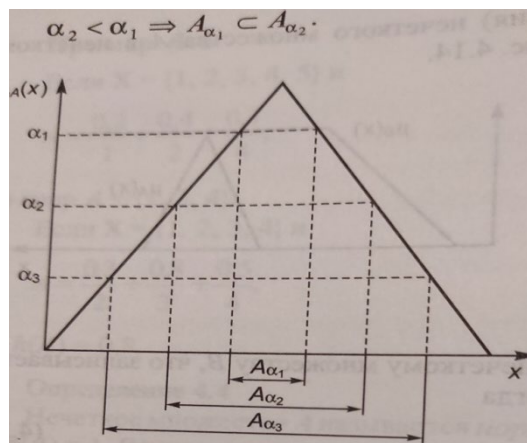


Рисунок 3.6 Схема зв'язку між областями можливих значень параметру «X» і  $\alpha$ -рівнями трикутних моделей нечітких величин  $\{S_1, \dots, S_5\}$ .

На рис. 3.6 приведена схема, яка демонструє перехід від області можливих значень параметру «X», які задаються при вводі даних параметрів  $\{\text{Мод } P_k\}$ , до значень відповідних  $\alpha$ -рівнів. Рис. 2 показує, що чим менший діапазон можливих значень, тим вищій  $\alpha$ -рівень для відповідних параметрів апроксимації  $S = \{S_1, S_2, \dots, S_5\}$ .

1. Введення параметрів  $\{\text{Мод } P_k\}$ :

Нечіткі величини (НВ) або коефіцієнти упевненості  $CF(A)$ .

2. Отримання "Е-еталону":

Вектор ознак моделі класифікації.

Задача - визначення найближчого шаблону  $\{\text{Мод } P_k\}$  до "Е-еталону".

3. Перший метод оцінювання (МО1):

Оцінка показника  $\{P_k\}$  безпосередньо в межах НВ або  $CF(A)$ .

4. Другий метод оцінювання (МО2):

Використання нечіткої моделі "Показник" (МПок).

Модель апроксимації області можливих оцінок НВ.

Формування значень на основі опитування експертів/менеджерів.

5. Апроксимація трикутними величинами  $S$ :

Область значень показника через трикутні величини  $S = \{S_1, S_2, \dots, S_5\}$ .

6. Формування  $\alpha$ -рівня на основі правил для  $S$ :

Розрахунок  $\alpha$ -рівня відповідно до наближеного діапазону показника "Т".

7. Перехід від області можливих значень до  $\alpha$ -рівнів:

Перехід від введених даних параметрів {Мод Pk} до відповідних значень  $\alpha$ -рівнів на основі діапазонів.

Приклад розробленої програми

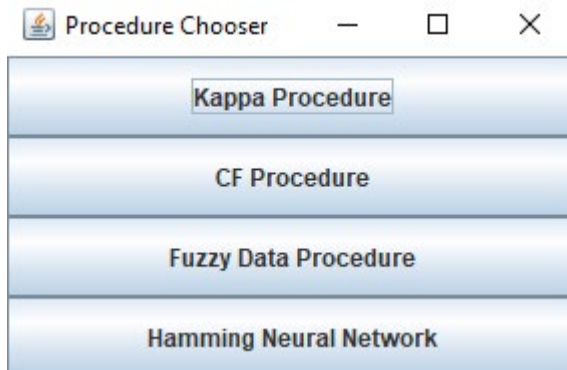


Рисунок 3.7 Вікно вибору процедури

Процедура Каппи Коена

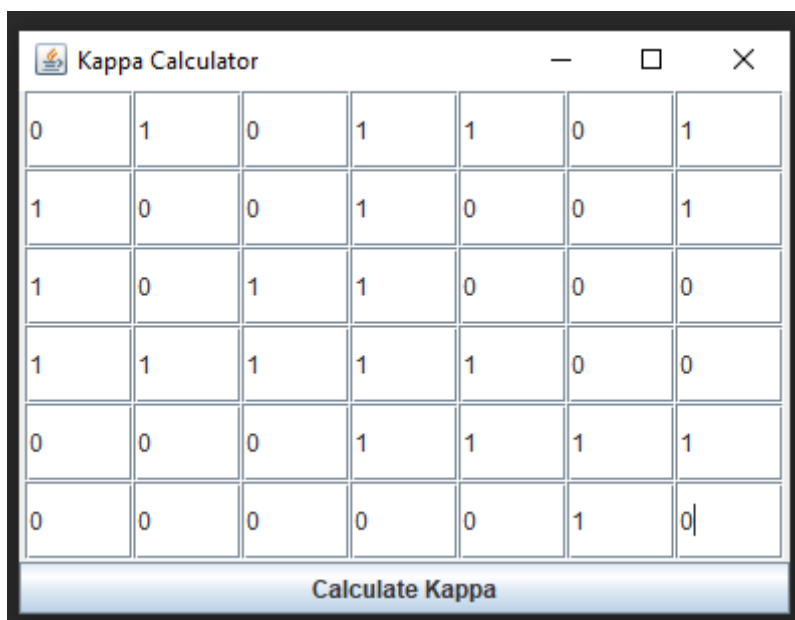


Рисунок 3.8 Вікно внесення даних процедури Каппи Коена

Після натиснення кнопки розрахунку з'являється вікно з результатом розрахунку

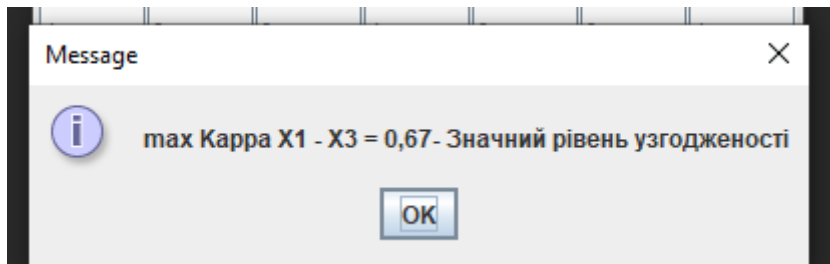


Рисунок 3.9 Вікно результату процедури Каппи Коена

Процедура CF (коефіцієнтів впевненості)

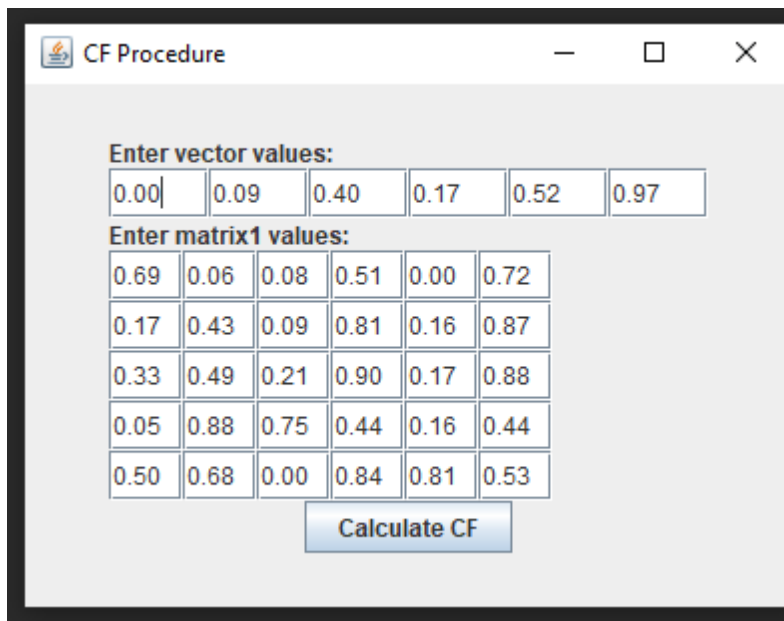


Рисунок 3.10 Вікно внесення даних процедури Коефіцієнти впевненості

Після натиснення кнопки розрахунку, виводить результат розрахунку стосовно вектора.

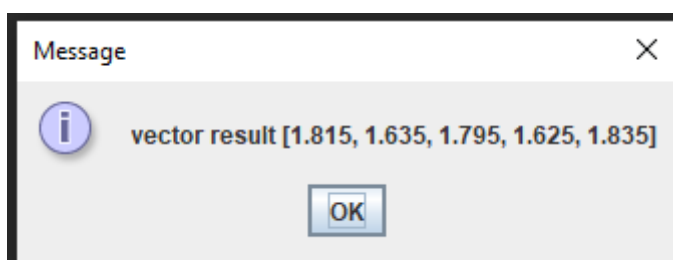


Рисунок 3.11 Вікно результату розрахунку вектора процедури Коефіцієнти впевненості

В іншому вікні відкривається результат розрахунку і знаходження відповідності за індексом відповідно до вхідних даних

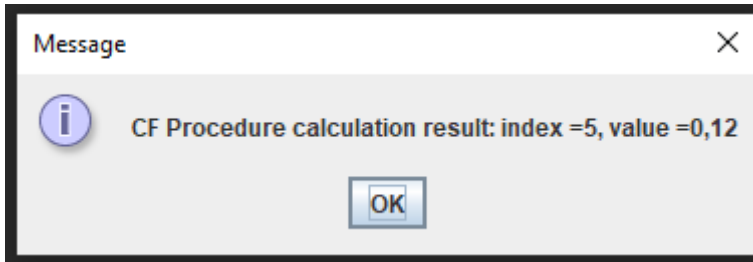


Рисунок 3.11 Вікно результату розрахунку Коефіцієнти впевненості

Процедура Fuzzy (нечіткості)

A screenshot of a software window titled "Fuzzy Procedure". It contains two sections for data entry:

**Enter vector values:**

0	1	0.3	0.5	0.5	0.03
---	---	-----	-----	-----	------

**Enter matrix1 values:**

0.15	0.72	0.34	0.38	0.89	0.89
0.21	0.28	0.28	0.89	0.72	0.99
0.70	0.63	0.89	0.72	0.39	0.63
0.75	0.21	0.75	0.63	0.28	0.72
0.89	0.75	0.21	0.99	0.39	0.99

At the bottom center of the window is a "Calculate Fuzzy" button.

Рисунок 3.12 Вікно внесення даних процедури Нечіткості

Після натиснення кнопки розрахунку, виводить результат розрахунку стосовно вектора.

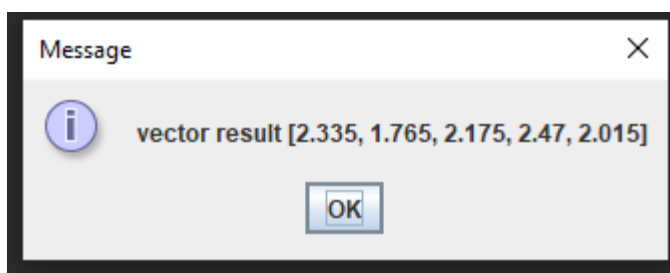


Рисунок 3.13 Вікно результату розрахунку вектора процедури Нечіткості

В іншому вікні відкривається результат розрахунку і знаходження відповідності за індексом відповідно до вхідних даних

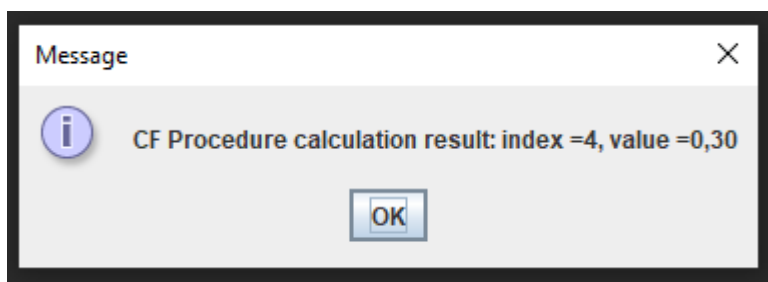


Рисунок 3.11 Вікно результату розрахунку процедури Нечіткості

## ВИСНОВКИ

У контексті управління проектами в умовах невизначеності та моделювання завдань менеджменту, розглянуті розділи вносять значущий внесок у розуміння та вдосконалення процесів управління проектами. Важливою частиною цього аналізу є розгляд стратегій, процедур та інструментів, які дозволяють ефективно керувати проектами в умовах невизначеності.

Аналіз цього розділу свідчить про те, що процедури мережі Хеммінга є потужним інструментом для класифікації недетермінованих та слабо структурованих даних. Виявлено, що їхня ефективність забезпечується агільним плануванням, гнучкістю в управлінні ризиками та ітераційним підходом, що робить їх відмінним інструментом для моделювання в умовах невизначеності.

Висвітлення характеристик сфери використання процедур управління проектами в умовах невизначеності підкреслює їхню ключову роль у стратегічному плануванні, гнучкості та адаптабельності, ефективному управлінні ризиками та залученні команди. Ці характеристики є важливими факторами успіху в управлінні проектами в невизначених умовах.

Аналіз результатів моделювання завдань менеджменту підкреслює значущість ефективних стратегій управління ризиками, гнучкості та адаптабельності, використання сучасних технологій та активного залучення команди в умовах невизначеності. Вивчення успіхів та помилок створює основу для подальшого вдосконалення стратегій та процедур управління проектами.

Загалом, результати аналізу свідчать про необхідність адаптації та оптимізації стратегій та процедур управління проектами в умовах невизначеності для досягнення успішних результатів та виконання проєктів з високою ефективністю.

## 4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСОБІВ МЕНЕДЖМЕНТУ УПРАВЛІННЯ ПРОЄКТАМИ ПРОГРАМНОЇ ІНЖЕНПЕРІЇ ЗА УМОВ НЕВИЗНАЧЕНОСТІ ДАНИХ

### 4.1 Дослідження процедур нечіткої класифікації та редукції моделей даних об'єктів.

Це завдання полягає в підвищенні достовірності моделей класифікації в умовах невизначених даних, які представлені у формі нечітких величин. Для досягнення цієї мети використовуються процедури редукції та статистики Коена, які спрямовані на оптимізацію точності та надійності класифікації в умовах невизначеності.

У контексті цього завдання визначаються постановки та моделі для класифікації за нечіткими даними. Ці моделі призначені для вибору кандидата на конкретну роль, враховуючи нечіткість представлених даних та вагомість впевненості, визначену за допомогою коефіцієнтів упевненості  $CF(A)$ .

Процедури редукції використовуються для спрощення та оптимізації нечітких величин, що дозволяє отримати більш точні та інтерпретовані результати класифікації. Статистика Коена використовується для оцінки ступеня узгодженості між різними класифікаційними моделями та визначення їхньої точності.

Отже, вирішення цього завдання включає розробку моделей, які забезпечують класифікацію за нечіткими даними, а також використання процедур редукції та статистики Коена для поліпшення точності та надійності класифікаційних результатів у ситуаціях невизначеності.

Начало формы

Задаємо вхідний вектор та значення

Fuzzy Procedure

Enter vector values:

0.16	0.97	0.41	0.41	0.27	0.67
------	------	------	------	------	------

Enter matrix1 values:

0.15	0.54	0.48	0.57	0.71	0.09
0.88	0.00	0.79	0.68	0.35	0.27
0.02	0.04	0.36	0.94	0.48	0.74
0.29	0.06	0.36	0.28	0.61	0.30
0.33	0.80	0.09	0.20	0.32	0.07

Calculate Fuzzy

Рисунок 4.1. – вікно внесення даних процедури нечіткості

При даних значеннях результуючий вектор дорівнює:

Message

vector result [2,16, 1,59, 2,03, 2,04, 2,24]

OK

Рисунок 4.2. – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 5, зі значенням 0.23

Message

Fuzzy Procedure calculation result: index =5, value =0,23

OK

Рисунок 4.3. – вікно результатів розрахунку процедури

2 варіант

Задаємо вхідний вектор та значення які дорівнюють 1

Enter vector values:

0.86	0.35	0.03	0.34	0.86	0.25
------	------	------	------	------	------

Enter matrix1 values:

0.86	0.35	0.03	0.34	0.86	0.25
0.42	0.41	0.18	0.00	0.62	0.67
0.11	0.18	0.61	0.90	0.84	0.33
0.80	0.90	0.76	0.05	0.29	0.90
0.04	0.96	0.79	0.32	0.47	0.15

Calculate Fuzzy

Рисунок 4.4. – вікно внесення даних процедури нечіткості

При даних значеннях результуючий вектор дорівнює:



Рисунок 4.5. – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 1, зі значенням 1.14

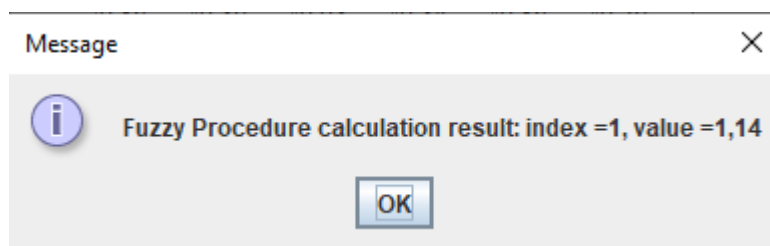


Рисунок 4.6. – вікно результатів розрахунку процедури

3 варіант

Задаємо вхідний вектор та значення які близькі до значень 3

Enter vector values:

0.1	0.2	0.6	1	0.8	0.3
-----	-----	-----	---	-----	-----

Enter matrix1 values:

0.86	0.35	0.03	0.34	0.86	0.25
0.42	0.41	0.18	0.00	0.62	0.67
0.11	0.18	0.61	0.90	0.84	0.33
0.80	0.90	0.76	0.05	0.29	0.90
0.04	0.96	0.79	0.32	0.47	0.15

Calculate Fuzzy

Рисунок 4.7. – вікно внесення даних процедури нечіткості

При даних значеннях результуючий вектор дорівнює:

Message

vector result [1,88, 1,75, 2,90, 1,19, 1,91]

OK

Рисунок 4.8. – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 3, зі значенням 1.21

Message

Fuzzy Procedure calculation result: index =3, value =1,21

OK

Рисунок 4.9. – вікно результатів розрахунку процедури

Це дослідження показує що модель працює достовірно.

## 4.2 Дослідження процедур класифікації на основі даних достовірності CF(X).

У даній роботі розглядається модель класифікації в контексті управлінського відбору кандидатів із зазначеної множини. Зазначене завдання призначення кандидатів відоме і має різноманітні моделі та процедури реалізації, залежно від різних параметрів, способів їх отримання та оцінювання тощо.

У викладеній моделі вважається, що процедура відбору кандидатів здійснюється на основі їхніх портфоліо, яке містить перелік виконаних робіт у конкретних проєктах або резюме. Кожен кандидат може мати різний набір завдань у своєму портфоліо, описаний без попередніх вимог до структури цих завдань.

Характеристики та оцінки для кожного завдання в портфоліо можуть бути представлені у текстовій або мовно-формальній формі та відрізнятися для різних кандидатів. Оцінки можуть також включати нечіткі показники. Остаточну структуру та значення оцінок для представлення завдань визначає менеджмент, який встановлює оцінки для всіх ознак у формі коефіцієнтів упевненості в контексті нечіткої логіки.

У підсумку, шаблони різних кандидатів у моделі неоднорідні, можуть містити "пропуски" в ознаках, і кількість шаблонів кандидатів може відрізнятися. Вхідний "еталон" включає всі характеристики, які враховуються та визначаються менеджментом при формуванні моделі відбору кандидатів за портфоліо. Основною метою є визначення шаблону та кандидата, який найбільш відповідає "еталону" у процесі вирішення подібних завдань управлінського відбору.

Задаємо вхідний вектор та значення

CF Procedure

Enter vector values:

-0.33	0.08	-0.27	0.17	-0.18	0.04
-------	------	-------	------	-------	------

Enter matrix1 values:

-0.34	-0.23	-0.12	0.54	0.53	0.96
-0.70	-0.60	0.68	0.92	0.16	0.77
-0.76	0.86	0.56	-0.58	-0.29	-0.47
-0.40	-0.38	0.28	-0.32	-0.17	0.15
-0.50	0.90	0.58	-0.30	0.97	-0.13

Calculate CF

Рисунок 4.10 – вікно внесення даних процедури CF

При даних значеннях результуючий вектор дорівнює:

Message

vector result [1,77, 1,09, 1,30, 2,16, 1,19]

OK

Рисунок 4.11 – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 4, зі значенням 0.71

Message

CF Procedure calculation result: index =4, value =0,71

OK

Рисунок 4.12 – вікно результатів розрахунку процедури

2 варіант

Задаємо вхідний вектор та значення які дорівнюють 5

Рисунок 4.13 – вікно внесення даних процедури CF

При даних значеннях результуючий вектор дорівнює:

Рисунок 4.14 – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 5, зі значенням 1.69

Рисунок 4.15 – вікно результатів розрахунку процедури

Результат показує що процедура працює відповідно до завдання.

3 варіант

Задаємо вхідний вектор та значення які близькі до значень 3

CF Procedure

Enter vector values:

-0.7	0.9	0.58	-0.5	-0.33	-0.5
------	-----	------	------	-------	------

Enter matrix1 values:

-0.34	-0.23	-0.12	0.54	0.53	0.96
-0.70	-0.60	0.68	0.92	0.16	0.77
-0.76	0.86	0.56	-0.58	-0.29	-0.47
-0.40	-0.38	0.28	-0.32	-0.17	0.15
-0.50	0.90	0.58	-0.30	0.97	-0.13

Calculate CF

Рисунок 4.16 – вікно внесення даних процедури CF

При даних значеннях результуючий вектор дорівнює:

Message

vector result [0,22, 0,61, 2,86, 1,57, 1,96]

OK

Рисунок 4.17 – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 3, зі значенням 1.53

Message

CF Procedure calculation result: index =3, value =1,53

OK

Рисунок 4.18 – вікно результатів розрахунку процедури

Це дослідження показує що модель працює достовірно.

### 4.3 Процедура Каппа Коена і редукції

При формуванні конкуруючих варіантів моделей класифікації (КМ) будемо розглядати результати для всіх комбінацій пар скорочених моделей  $X/\{X_j\}$ , а також наступні передумови аналізу результатів. КМ1 – оцінюється класифікація хоча б одним із шаблонів  $L_{ij}$  класу  $K_i$ , КМ2 – оцінюється класифікація кожним із  $L_{ij}$ . Також досліджуються різні способи формування таблиць розбіжностей для розрахунку показників каппа (1). Відповідно першого способу, С1, враховуються відомі попередні результати класифікації за набором  $X$ , відповідно способу С2 такі дані не використовуються, контролюються лише отримані результати (шаблони/класи) за МХН.

Таблиця 4.1

Результати циклу процедури редукції при формуванні моделі класифікації

$K_i$	$L_i$	$X$	$X/\{X_1\}$	$X/\{X_2\}$	$X/\{X_3\}$	$X/\{X_4\}$	$X/\{X_5\}$	$X/\{X_6\}$
K1	L11		+					+
	L12	+		+		+		
	L13				+			+
K2	L21					+		+
	L22		+	+				
	L23				+			
	L24	+				+		
K3	L31		+	+				
	L32	+					+	
K4	L41							
	L42		+		+	+		
	L43							
	L44	+				+		+
	L45					+		
K5	L51			+				
	L52						+	
	L53							
K6	L61			+				
	L62				+	+		
	L63	+						
	L64				+			
K7	L71						+	

	L72				+			
	L73	+						+
K8	L81					+		
	L82	+	+	+				

Результати формування таблиць розбіжностей і розрахунків показників каппа (1) за даними табл. 1 для моделі класифікації KM1, коли в якості конкуруючих варіантів розглядалися всі комбінації пар скорочених моделей  $X/\{X_j\}$ , наступні.

Таблиця 4.2

Розрахунки показників каппа статистики за таблицями розбіжностей

Розрахунок за класами ( KM1, C2)

Розрахунок за класами (KM1, C1)

$$K(X1/X2) = 0,143$$

	Так	ні
так	4	1
ні	2	1

$$K(X1/X2) = 0,143$$

	Так	ні
так	4	2
ні	1	1

$$K(X1/X3) = -0,067$$

	Так	ні
так	3	2
ні	2	1

$$K(X1/X3) = -0,333$$

	Так	ні
так	4	2
ні	2	0

$$K(X1/X4) = 0,467$$

	Так	ні
так	4	1
ні	1	2

$$K(X1/X4) = 0,333$$

	Так	ні
так	5	1
ні	1	1

$$K(X1/X5) = -0,412$$

	Так	ні
так	1	4
ні	2	1

$$K(X1/X5) = -0,2$$

	Так	ні
так	1	5
ні	1	1

$$K(X1/X6) = 0,25$$

	Так	ні
так	3	2
ні	1	2

$$K(X1/X6) = 0,143$$

	Так	ні
так	4	1
ні	2	1

$$K(X2/X3) = -0,429$$

	Так	ні
так	3	3
ні	2	0

$$K(X2/X3) = -0,333$$

	Так	ні
так	4	2
ні	2	0

$$K(X2/X4) = 0,143$$

	Так	ні
--	-----	----

$$K(X2/X4) = 0,143$$

	Так	ні
--	-----	----

	так	4	2		так	4	1
	ні	1	1		ні	2	1
$K(X2/X5) = -0,111$		Так	ні	$K(X2/X5) = -0,111$		Так	ні
	так	2	4		так	1	4
	ні	1	1		ні	1	2
$K(X2/X6) = -0,500$		Так	ні	$K(X2/X6) = -0,600$		Так	ні
	так	2	4		так	1	4
	ні	1	1		ні	1	2
$K(X3/X4) = 0,467$		Так	ні	$K(X3/X4) = 0,333$		Так	ні
	так	4	1		так	5	1
	ні	1	2		ні	1	1
$K(X3/X5) = -0,412$		Так	ні	$K(X3/X5) = -0,200$		Так	ні
	так	1	4		так	1	5
	ні	2	1		ні	1	1
$K(X3/X6) = 0,750$		Так	ні	$K(X3/X6) = 0,714$		Так	ні
	так	4	1		так	5	1
	ні	0	3		ні	0	2
$K(X4/X5) = -0,882$		Так	ні	$K(X4/X5) = -0,600$		Так	ні
	так	3	2		так	0	6
	ні	1	2		ні	2	0
$K(X4/X6) = 0,250$		Так	ні	$K(X4/X6) = 0,143$		Так	ні
	так	0	5		так	4	2
	ні	3	0		ні	1	1

У табл. 4.2 приведена частина результатів розрахунків оцінок «каппа», яка дозволяє визначити першу змінну для скорочення моделі класифікації –  $X_6$ ,  $K(X_3/X_6) = 0,75$  (значний рівень узгодженості). Для інших не приведених у табл. 2 пар оцінки величин «каппа» були несуттєвими, подібність відсутня, як для  $K(X_1/X_3) = -0,067$ . Можна сказати, що «вплив» фактору  $X_6$  на модель класифікації ураховують та опосередковано представляють фактори  $X_1$ ,  $X_3$  і  $X_4$ . Відзначається, що оцінки подібності за

каппа статистикою для таблиць розбіжностей без урахування класифікації за набором  $X$ , (спосіб  $C2$ ) перевищують чи дорівнюють результати за способом  $C1$ . Приведемо у табл. 3 основні результати розрахунків оцінок показників «каппа» для скороченого складу параметрів класифікації,  $(X_1, X_2, \dots, X_5)$ , представлені за схемою табл. 2.

Таблиця 4.3

$K(X_1/X_2) = -0,600$		Так	ні	$K(X_1/X_3) = 0,529$		Так	ні
	так	2	3		так	3	2
	ні	3	0		ні	0	3
$K(X_2/X_5) = -0,412$		Так	ні	$K(X_3/X_4) = 0,467$		Так	ні
	так	1	4		так	2	1
	ні	2	1		ні	1	4
$K(X_3/X_5) = 0,143$		Так	ні	$K(X_4/X_5) = -0,429$		Так	ні
	так	1	2		так	0	3
	ні	1	4		ні	2	3

Найвище значення «каппа» Коена  $K(X_1/X_3) = 0,53$  (помірний рівень узгодженості), тому видаляється  $X_3$  (подібність до  $X_1, X_4, X_5$ ). Відзначається також відмінність результатів стосовно етапів табл. 2 та табл. 3 за способами формування таблиць розбіжностей  $C1$  і  $C2$ . А саме – щодо врахування попередньо відомих результати класифікації за набором  $X$ . У табл. 3 більш високі оцінки показників були отримані за способом  $C1$ , а не табл. 2. Тобто способи  $C1$  та  $C2$  необхідно розглядати обидва при виконанні скорочення числа параметрів моделі класифікації, обираючи рішення за більшими значеннями «каппа». Зауважимо про необхідність перевірки тотожності та можливості корегування набору шаблонів моделі класифікації після видалення параметрів  $\{X_j\}$ . У разі утворення кількох однакових шаблонів скороченої моделі  $X/\{X_j\}$  у одному або у кількох класах  $K_i$ , залишається лише один, щоб не впливати на значення оцінок «каппа» (1) на наступних етапах процедури. Якщо такі шаблони виникли в одному класі, треба залишити тільки один, а в разі належності зазначених зразків до різних класів  $K_i$  видаляються шаблони із класу з більшим числом екземплярів.

Наведемо результати процедури редукції для КМ2, коли оцінюється класифікація табл. 1 кожним із 26 зразків  $L_{ij}$ .

Таблиця 4.4

Розрахунки при  $m=6$  за зразками (КМ2, С2)      Розрахунки при  $m=6$  за зразками (КМ2, С1)

$K(X1/X2) = 0,425$		Так	ні	$K(X1/X2) = 0,494$		Так	ні
	так	3	2		так	14	4
	ні	3	18		ні	2	6
$K(X1/X3) = -0,035$		Так	ні	$K(X1/X3) = 0,299$		Так	ні
	так	1	4		так	10	5
	ні	5	16		ні	4	7
$K(X1/X5) = -0,169$		Так	ні	$K(X1/X5) = 0,325$		Так	ні
	так	0	5		так	13	5
	ні	3	18		ні	3	5
$K(X3/X5) = -0,182$		Так	ні	$K(X3/X5) = 0,308$		Так	ні
	так	0	6		так	11	2
	ні	3	17		ні	7	6
$K(X5/X6) = -0,169$		Так	ні	$K(X5/X6) = 0,278$		Так	ні
	так	0	3		так	14	4
	ні	5	18		ні	4	4

Параметр  $X1$  табл. 4 має найвищий показник «каппа»  $K(X1/X2)=0,494$  (помірний рівень узгодженості) і разом з тим високі оцінки подібності для  $K(X1/X3) = 0,299$  та  $K(X1/X5) = 0,325$ , які перевищують значення інших.  $X1$  представлений через зазначені змінні, тому  $X1$  видаляється з моделі. У табл. 4 також більш високі оцінки показників «каппа» отримані за способом С1.

Таблиця 4.5

Розрахунки при  $m=5$  за зразками з урахуванням даних  $X$  (КМ2, С1)

$K(X2/X3) = 0,150$		Так	ні	$K(X2/X4) = 0,103$		Та к	ні
	так	12	5		так	13	4
	ні	5	4		ні	6	3

$K(X3/X4) = 0,103$		Так	Ні
	так	13	4
	ні	6	3

$K(X3/X5) = 0,320$		Так	Ні
	так	13	4
	ні	4	5

$K(X3/X6) = 0,324$		Так	Ні
	так	10	7
	ні	2	7

$K(X4/X5) = 0,283$		Так	Ні
	так	14	5
	ні	3	4

$K(X4/X6) = 0,331$		Так	Ні
	так	11	8
	ні	1	6

$K(X5/X6) = 0,173$		Так	Ні
	так	9	8
	ні	3	6

Найвища оцінка коефіцієнту «каппа» у пари  $K(X4/X6) = 0,331$  (помірний рівень узгодженості), а  $K(X3/X6) = 0,324$  і  $K(X5/X6) = 0,173$  - видаляється  $X6$ .

#### 4.4 Дослідження достовірності процедур нечіткої класифікації щодо завдань менеджменту управління проєктами за умов невизначеності .

Задаємо вхідний вектор та значення

HammingNeuralNetwork

Enter vector values:

0.48	0.54	0.12	0.44	0.03	0.50
------	------	------	------	------	------

Enter matrix1 values:

0.84	0.06	0.33	0.54	0.02	0.57
0.06	0.82	0.21	0.91	0.87	0.44
0.55	0.87	0.95	0.75	0.14	0.42
0.74	0.63	0.90	0.51	0.14	0.75
0.55	0.60	0.60	0.68	0.29	0.55

Calculate

Рисунок 4.19 – вікно внесення даних процедури Хеммінга

При даних значеннях результуючий вектор дорівнює:

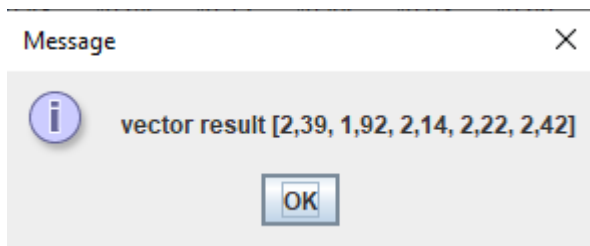


Рисунок 4.20 – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 5, зі значенням 0.16

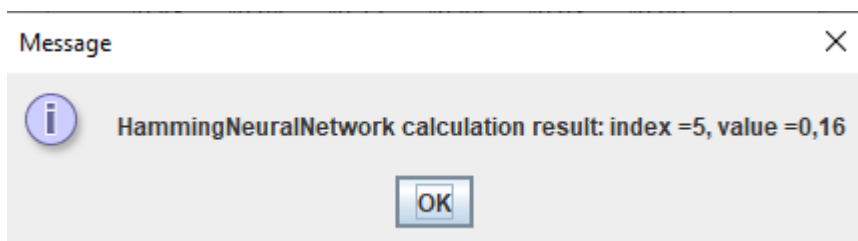


Рисунок 4.21 – вікно результатів розрахунку процедури

2 варіант

Задаємо вхідний вектор та значення які дорівнюють 1

A screenshot of the "HammingNeuralNetwork" application window. It features two input sections: "Enter vector values:" and "Enter matrix1 values:". The vector input contains a row of six text boxes with values 0.55, 0.87, 0.95, 0.75, 0.14, and 0.42. The matrix input contains a 5x6 grid of text boxes with the following values:

0.84	0.06	0.33	0.54	0.02	0.57
0.06	0.82	0.21	0.91	0.87	0.44
0.55	0.87	0.95	0.75	0.14	0.42
0.74	0.63	0.90	0.51	0.14	0.75
0.55	0.60	0.60	0.68	0.29	0.55

Below the matrix grid is a "Calculate" button.

Рисунок 4.22 – вікно внесення даних процедури Хеммінга

При даних значеннях результуючий вектор дорівнює:



Рисунок 4.23 – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 1, зі значенням 1.14

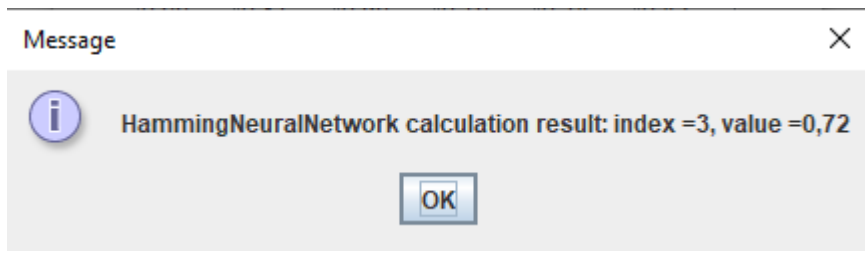


Рисунок 4.24 – вікно результатів розрахунку процедури

3 варіант

Задаємо вхідний вектор та значення які близькі до значень 3

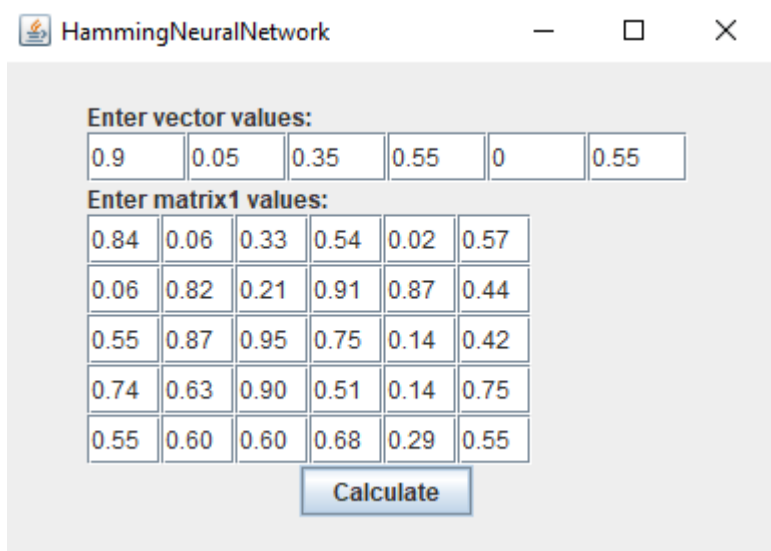


Рисунок 4.25 – вікно внесення даних процедури Хеммінга

При даних значеннях результуючий вектор дорівнює:

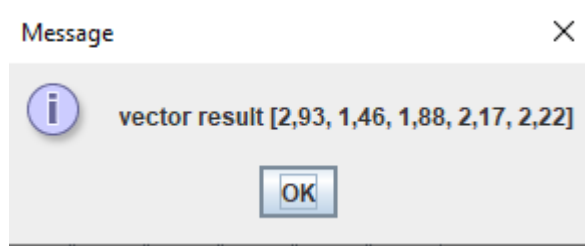


Рисунок 4.26 – вікно результатів розрахунку вектора

Було знадено відповідність до індексу 3, зі значенням 1.21

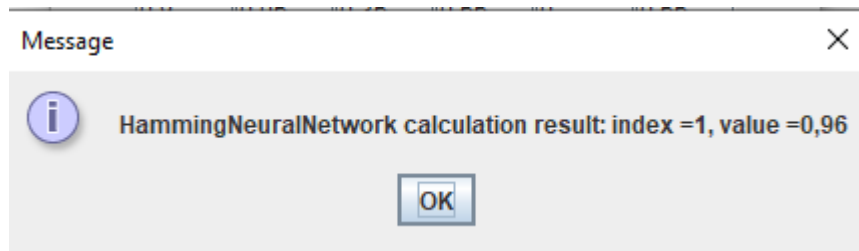


Рисунок 4.27 – вікно результатів розрахунку процедури

Це дослідження показує що модель працює достовірно.

## ВИСНОВКИ

У роботі виконано дослідження та отримано результати щодо формування інтелектуальних програмних засобів моделювання процедур менеджменту програмної інженерії при управлінні проектами. При тому в якості об'єкту дослідження використовувалися методи та засоби моделювання процедур менеджменту програмної інженерії при управлінні проектами. Математичні моделі та алгоритми процедур менеджменту програмної інженерії при управлінні проектами, процедури асоціативної пам'яті мережі Хеммінга, а також процедури редукції математичних моделей і статистики «каппа Коена» являлися предметом дослідження дипломної роботи.

Мета роботи була у розвитку постановок завдань та удосконалення математичних моделей для завдань менеджменту програмної інженерії при управлінні проектами. Разом з цим виконана реалізація завдань відтворення достовірних багатопараметричних моделей класифікації (ДБМК) при невизначених даних на основі застосування процедур редукції та каппа статистики. В якості характеристик моделей ДБМК були обрані структури параметрів (кількість і склад), тип даних і множина шаблонів, які складають базу даних/знань досліджуваної предметної області.

Дослідження виконувалися такими методами: методи порівняльного аналізу, статистики, класифікації багатопараметричних об'єктів, методи нечіткого моделювання, методи експериментальних досліджень та комп'ютерне моделювання. Для проектування та розробки програми використані методи програмної інженерії використовувалися.

Новизна отриманих результатів визначається таким:

- виконано аналіз моделей та процедур широкого кола завдань менеджменту програмної інженерії при управлінні проектами,

- запропоновано нові постановки завдань призначення на посаду (вибору кандидата), класифікації за нечіткими даними та ознаками у формі коефіцієнтів упевненості,
- для реалізації завдань класифікації при невизначених даних модифіковані моделі асоціативної пам'яті Хеммінга,
- для забезпечення результатів класифікації з встановленими ймовірнісними показниками розроблено процедуру редукції розмірності математичної моделі з використанням статистики «капа Коена»,
- проведений широкий числовий експеримент підтвердив достовірність та ефективність запропонованих моделей і методів.

В роботі отримано програмну реалізацію розроблених моделей класифікації для менеджменту програмної інженерії при управлінні проектами з неточно визначеними параметрами.

Розрахунково-пояснювальна записка складається зі вступу, 4 розділів, висновків, бібліографічного списку та додатків.

Вступ – представляє сутність завдань моделювання процедур менеджменту програмної інженерії при управлінні проектами при неточно визначених характеристик даних..

Перший розділ – містить опис матеріалів наукових літературних джерел щодо сутності та результатів завдань моделювання процедур менеджменту програмної інженерії при управлінні проектами при неточно визначених характеристик даних,

Другий розділ – містить результати розвитку постановок завдань менеджменту програмної інженерії при управлінні проектами на основі інтелектуальних процедур при не точно визначених характеристиках даних, моделі завдань призначення за умов неточно визначених даних, результати модифікування моделей нейронних мереж Хеммінга, нечіткі та форми представлення неточно визначених даних у вигляді коефіцієнтів упевненості, процедуру редукції математичних моделей завдань класифікації на основі

статистики «каппа Коена», а також результати чисельних досліджень запропонованих алгоритмів та процедур.

Третій розділ – предствіляє формування структури програмного комплексу щодо завдань менеджменту програмної інженерії при управлінні проектами, результати проектування і розробки програми.

Четвертий розділ – містить результати експериментальних досліджень можливостей запропонованих процедур програмного комплексу, які підтвердили достовірність та ефективність результатів досліджень та розробок. .

Практичне значення мають результати щодо

- нових постановок завдань із призначення на посаду (вибору кандидата), класифікації за нечіткими даними та ознаками у формі коефіцієнтів упевненості.

- модифіковані моделі асоціативної пам'яті Хеммінга,

- процедура редукції розмірності математичної моделі з використанням статистики «капа Коена»,

- позитивні результати проведеного числового експерименту, що підтвердили достовірність та ефективність запропонованих моделей і методів

Результати дослідження і розробок пройшли апробацію, вони доповідались на семінарах кафедри КІТ 11.01.2024 р. Матеріали роботи доповідалися на 17 міжнародній науково-практичній конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» (13.12.2023 – 14.12.2023 р.) м. Дніпро, Український державний університет науки і технологій

Результати досліджень та розробок опубліковані у фаховій науковій статі (додаток).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скалозуб В. В., Горячкін В. М., Клименко І. В., Терлецький І. А., Терленко А. П. Дослідження процедур мережі хеммінга для управління сервісними системами при неточно визначених і природомовних даних. *Наука та прогрес транспорту*. 2022. № 3-4 (99-100). С. 33–47. DOI: <https://doi.org/10.15802/stp2022/276411>
2. Великоіваненко Г. І. *Оцінювання рівня економічної безпеки на підґрунті відстані Хеммінга*. 2018. URL: <https://core.ac.uk/download/pdf/197269753.pdf>
3. Васильев В. И. Индукция и редукция в проблемах экстраполяции. *Кибернетика и вычислительная техника*. 1998. Вып. 116. С. 65–81.
4. Freitag R. M. Ко. Kappa statistic for judgment agreement in Sociolinguistics / Estatística Kappa para concordância de julgamento em Sociolinguística. *Revista de Estudos da Linguagem*. 2019. Vol. 27, No. 4. P. 1591–1612. DOI: <https://doi.org/10.17851/2237-2083.0.0.1591-1612>
5. Колесник А. С., Хайрова Н. Ф. Обґрунтування використання статистики каппа Коена в експериментальних дослідженнях NLP Text Mining. *Кибернетика та системний аналіз*. Т. 58, № 2. 2022. С. 143–153.
6. Li Min Fu, Shortliffe E. H. The application of certainty factors to neural computing for rule discovery. *IEEE Transactions on Neural Networks*. 2000. Vol. 11. Iss. 3. P. 647–657. DOI: <https://doi.org/10.1109/72.846736>
7. Скалозуб В. В., Горячкін В. М., Терлецький І. А. БАГАТОПАРАМЕТРИЧНІ ІНТЕЛЕКТУАЛЬНІ ПРОЦЕДУРИ ДІАГНОСТУВАННЯ ЗА НЕПОВНИМИ ТА ЗБУРЕНИМИ ДАНИМИ [Текст]: матеріали доповідей II Міжнародної науково-практичної конференції ЛОГІСТИКА І ТРАНСПОРТНА БЕЗПЕКА: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ РОЗВИТКУ В КОНТЕКСТІ АНАЛІЗУ СУЧАСНИХ ВИКЛИКІВ І ЗАГРОЗ, 09 листопада 2023 р. — Дніпро: Середняк Т.К. С. 42 – 48.
8. Скалозуб В.В., Горячкін В.М., Терлецький І.А. Інтелектуальна технологія оптимізації керування потоками замовлень сервісних систем із неточно визначеними і природномовними даними. *Наука та прогрес транспорту*, 2023, № 2 (102). – С. 54 – 70. doi: <https://doi.org/10.15802/stp2023/288077>
9. Еш С. І. Індуктивні методи визначення закономірностей *Кибернетика та обчислювальна техніка*. 2003. Вип. 138. С. 33–41.
10. Шинкаренко В.І., Демидович І.М. Визначення ознак авторства природномовних текстів //Штучний інтелект. № 3. 2018, С. 27-35.
11. Shynkarenko V.I., Demidovich I.M. Authorship Determination of Natural Language Texts by Several Classes of Indicators with Customizable Weights, in: Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021). Volume I: Main Conference. Lviv, Ukraine, April 22-23, 2021, pp. 832-844.
12. S.Min, H.Lee, and D. Bae, SoftPM: A Software Process Management System Reconciling Formalism with Easiness, *Information and Software Technology*, 42(1), 2000, pp. 1-16.

13. Young, R., & Jordan, E. (2008). Top management support: Mantra or necessity? *International Journal of Project Management*, 26(7), 713– 725
14. Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. In *Proceedings - 2014 Agile Conference, AGILE 2014* (Vol. 81, pp. 961– 971).
15. Ruhe, G., & Wohlin, C. (2014). Software project management in a changing world. In *Proceedings - 2014 Agile Conference, AGILE 2014* (Vol. 9783642550, pp. 1–477)
16. Caniëls, M.C.J., Bakens, R.J.J.M., 2012. The effects of Project Management Information Systems on decision making in a multi project environment, *International Journal of Project Management* 30, p. 162-175
17. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fifth Edition, 2013, Project Management Institute, Four Campus Boulevard, Newtown Square, PA 19073-3299 USA, p. 616
18. Rose, J., Pedersen, K., Hosband, J-H, Kraemmergaard, P. (2007) Management competences, not tools and techniques: A grounded examination of software project management at WM-data, *Information and Software Technol.* 49, 6 (Jun. 2007), 605-624.
19. Mishra, D., and Mishra, A. 2011. Complex software project development: agile methods adoption. *Journal of Software Maintenance and Evolution: Research and Practice*, 23(8), 549-564.
20. Lam, H.E. and Maheshwari, P. (2001) Task and Team Management in the Distributed Software Project Management Tool. In *Proceedings of the 25th international Computer Software and Applications Conference on invigorating Software Development* (October 08 - 12, 2001). COMPSAC. IEEE Computer Society, Washington, DC, 401-408.
21. Taherdoost, H. and Keshavarzsaleh, A. (2018) Managing Successful IT Project; Marketing Perspective. *Recent Advances on Computational Science and Applications*, 144-153.
22. Varajão, J. (2016) Success Management as a PM Knowledge Area—Work-in-Progress. *Procedia Computer Science*, 100, 1095-1102.
23. Asad, F. and Pinnington, A.H. (2013) Exploring the Value of Project Management: Linking Project Management Performance and Project Success. *International Journal of Project Management*, 32, 202-217.
24. Peters, L. and Moreno, A.M. (2017) Evaluating Software Project Managers: A Multidimensional Perspective. *IEEE Software*, 34, 104-108.
25. Dey, P.P., Amin, M., Sinha, B.R., Any, L. and Al Badkoobehi, H. (2018) Current Trends in Software Project Management. *Advances in Social Science, Education and Humanities Research*, 120, 22-26.

26. Cerdeiral, C.T. and Santos, G. (2018) Software Project Management in High Maturity: A Systematic Literature Mapping. *Journal of Systems and Software*, 148, 56-87.
27. Nakigudde, S. (2019) Project Management Models and Software Development Project Success. Master of Science in Information Systems, Makerere University.
28. Ackah, D. (2019) Project Management Methods, Methodologies, and Frameworks: An Exploration for Study Guild for Project Management Practitioners of Ghana. *Project Management Scientific Journal*, 1, 61-66.
29. Cullen, C.D. (2011) The Effectiveness of Software Project Management Practices: A Quantitative Measurement. Master's Thesis of Science in Software Engineering, Naval Postgraduate School, California, USA.
30. Foote, A. and Halawi, L.A. (2016) Knowledge Management Models within Information Technology Projects. *Journal of Computer Information Systems*, 58, 89-97.  
<https://doi.org/10.1080/08874417.2016.1198941>
31. Скалозуб В. В., Горячкін В. М., Клименко І. В., Терлецький І. А., Терленко А. П. Дослідження процедур мережі Хеммінга для управління сервісними системами при неточно визначених і природомовних даних. *Наука та прогрес транспорту*. 2022. № 3-4 (99-100). С. 33–47. DOI: <https://doi.org/10.15802/stp2022/276411> (in Ukrainian)
32. Великоіваненко Г. І. Оцінювання рівня економічної безпеки на підґрунті відстані Хеммінга. 2018. URL: <https://core.ac.uk/download/pdf/197269753.pdf>
33. Колесник А. С., Хайрова Н. Ф. Обґрунтування використання статистики каппа Коена в експериментальних дослідженнях NLP Text Mining. *Кібернетика та системний аналіз*. Т. 58, № 2. 2022. С. 143–153.
34. Li Min Fu, Shortliffe E. H. The application of certainty factors to neural computing for rule discovery. *IEEE Transactions on Neural Networks*. 2000. Vol. 11. Iss. 3. P. 647–657. DOI: <https://doi.org/10.1109/72.846736>
35. Скалозуб В. В., Горячкін В. М., Терлецький І. А. Багатопараметричні інтелектуальні процедури діагностування за неповними і збуреними даними // *Логістика і транспортна безпека: Проблеми та перспективи розвитку в контексті аналізу сучасних викликів і загроз: матеріали доповідей II Міжнародної науково-практичної конференції*. — Дніпро: Середняк Т.К., 2023. С. 42 – 47.
36. Шинкаренко В. І., Демидович І. М. Визначення ознак авторства природномовних текстів. *Штучний інтелект*. 2018. № 3. С. 27–35.
37. Richard A. Brualdi. *Combinatorial matrix classes*. — Cambridge: Cambridge University Press, 2006. — (Encyclopedia of Mathematics and Its Applications). — ISBN 0-521-86565-4

38. Leszek Rutkowski Metody i techniki sztucznej inteligencji. Naukowe PWN, Warszawa, 2005. – 520 p.
39. Haykin S. Neural networks: A Comprehensive Foundation. Prentice hall: New Jersey, 1999. 1103 p.

**ДОДАТКИ  
ДОДАТОК А**

**ЗАТВЕРДЖУЮ**

**Проректор  
Українського державного  
університету науки і  
технології  
Анатолій РАДКЕВИЧ**

**Технічне завдання  
ЛИСТ ЗАТВЕРДЖЕННЯ  
44165850.01345-01-ЛЗ**

**Завідувач кафедри КІТ**

\_\_\_\_\_ **Володимир ГОРЯЧКІН**

**Керівник розробки**

\_\_\_\_\_ **Владислав СКАЛОЗУБ**

**Виконавець**

\_\_\_\_\_ **Ілля ДУДНИК**

**Нормоконтролер**

\_\_\_\_\_ **Світлана ВОЛКОВА**

ЗАТВЕРДЖЕНО  
44165850.01345-01

Дослідження і моделювання процедур менеджменту програмної інженерії  
при управлінні проектами  
Технічне завдання

Листів 6

2024

## АНОТАЦІЯ

Документ 44165850.01345-01 «Дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проектами» Технічне завдання, що входить до складу програмної документації до дипломного проекту.

Цей документ визначає мету та сферу застосування програми, встановлює основні вимоги до неї, надає інформацію про етапи та терміни виконання проекту. Крім того, в програмній документації розглядаються технічні аспекти програми, а також її техніко-економічні показники. Це включає в себе технічні характеристики, методи розробки, тести, аналіз ефективності та вартість реалізації проекту. Програмна документація до дипломного проекту включає в себе детальний опис та інформацію про розроблену програму.

## ВСТУП

Програмний продукт для “ Дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проєктами ”, це інноваційне програмне забезпечення, розроблене для оптимізації управління проєктами в умовах невизначеності та змін. Завдяки своїм унікальним можливостям, воно призначене для:

Ітераційного Підходу:

Забезпечення можливості використання ітераційних методів для постійної адаптації стратегій та планів.

Використання Сучасних Технологій:

Інтеграція інноваційних технологій для збору та аналізу даних, спрямованих на управління невизначеністю.

ПЗ створено для того, щоб допомагати організаціям управляти своїми проєктами ефективно, навіть в умовах невизначеності, та досягати високих показників успішності в управлінні проєктами.

## **1 ПІДСТАВА ДЛЯ РОЗРОБКИ**

Основою для розробки є наказ ректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проектів» №1196 ст від 05.12. 2022 року.

Тема проекту: «Дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проектами».

Керівник дипломного проекту: Скалозуб В.В.

## **2 ПРИЗНАЧЕННЯ РОЗРОБКИ**

Функціональне призначення — Дослідження і моделювання процедур менеджменту програмної інженерії при управлінні проектами.

Експлуатаційне призначення — оптимізація та поліпшення ефективності управління проектами в умовах невизначеності та змін в бізнес-середовищі.

## **3 ВИМОГИ ДО ПРОГРАМИ**

### **3.1 Вимоги до функціональних характеристик**

Карра:

Опис функціональності:

Реалізація алгоритму Карра для визначення узгодженості між анотаторами.

Визначення ступеня узгодженості за допомогою специфікованих метрик.

CFProcedureGUI:

Опис функціональності:

Введення вектора та матриці за допомогою GUI.

Розрахунок результату за власною CF процедурою.

Виведення результатів у вікно сповіщень.

Нечіткі дані:

Опис функціональності:

Реалізація алгоритмів роботи з нечіткими даними.

Підтримка нечіткої логіки та операцій над нечіткими множинами.

Виведення результатів операцій над нечіткими даними.

### **3.2 Вимоги до надійності**

Стійкість до помилок:

Обробка помилок при введенні некоректних даних або відсутності даних.

### **3.3 Умови експлуатації**

Сумісність:

Підтримка роботи на різних платформах та середовищах.

Інтерфейс:

Зручний та інтуїтивний інтерфейс для користувачів.

Сумісність із стандартами GUI.

Застосування:

Можливість використання для обробки нечітких даних у різних областях.

### **3.4 Вимоги до складу і параметрів технічних засобів**

Мінімальні вимоги до обсягу оперативної пам'яті та процесорної потужності

### **3.5 Вимоги до маркування і упаковки**

Доступність документації, включаючи пояснення методів роботи та використання

Забезпечення належної упаковки програмного продукту для зручного розгортання та використання.

### **3.6 Вимоги до транспортування і зберігання**

Можливість зберігання результатів обчислень для подальшого використання.

Можливість перенесення програми між різними системами.

**ДОДАТОК Б****ЗАТВЕРДЖУЮ**

Проректор

Українського державного

університету науки і технології

Анатолій РАДКЕВИЧ

**ДОСЛІДЖЕННЯ І МОДЕЛЮВАННЯ ПРОЦЕДУР МЕНЕДЖМЕНТУ  
ПРОГРАМНОЇ ІНЖЕНЕРІЇ ПРИ УПРАВЛІННІ ПРОЄКТАМИ**

Керівництво користувача

ЛИСТ ЗАТВЕРДЖЕННЯ

44165850.01345-01 ІЗ 01

Завідувач кафедри КІТ

\_\_\_\_\_Вадим ГОРЯЧКІН

Керівник розробки

\_\_\_\_\_ ВЛАДИСЛАВ СКАЛОЗУБ

Виконавець

\_\_\_\_\_ Ілля ДУДНИК

Нормоконтролер

\_\_\_\_\_Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО

44165850.01345-01 ІЗ 01

ДОСЛІДЖЕННЯ І МОДЕЛЮВАННЯ ПРОЦЕДУР МЕНЕДЖМЕНТУ  
ПРОГРАМНОЇ ІНЖЕНЕРІЇ ПРИ УПРАВЛІННІ ПРОЄКТАМИ

Керівництво користувача  
44165850.01345-01 ІЗ 01

Листів 10

2024

### Анотація

Документ 44165850.01345-01 ІЗ 01 «Дослідження і моделювання процедур менеджменту. Керівництво користувача» входить до складу програмної документації на програму, що реалізує алгоритм редукції на основі каппа статистики виконується таким чином.

У даному документі представлено керівництво користувача. Програма написана на мові Java.

Програма не потребує конкретної ОС, вимагає лише наявної версії JDK.

## 1. Введення

В сучасному світі, де роль програмного забезпечення надто важлива для успішного функціонування підприємств і організацій, ефективне управління проектами в галузі програмної інженерії стає вирішальним фактором. Для досягнення успіху в цьому напрямку, важливо вивчати, досліджувати та моделювати процедури менеджменту програмної інженерії, які забезпечують оптимальну координацію ресурсів, контроль за термінами та якістю розробки.

Дослідження в цій області дозволяє розкрити ключові аспекти ефективного управління проектами в програмному секторі, включаючи в себе питання планування, контролю, комунікації та ризик-менеджменту. Моделювання процедур менеджменту стає потужним інструментом для аналізу та оптимізації різних сценаріїв управління проектами, дозволяючи виявити оптимальні стратегії та підходи.

Ця тема є актуальною, оскільки високотехнологічне середовище вимагає постійного вдосконалення методів та підходів до управління проектами в програмній інженерії. Дослідження та моделювання процедур менеджменту дозволяє розширити наше розуміння оптимальних стратегій управління проектами, що в свою чергу сприяє підвищенню ефективності розробки програмного забезпечення та досягненню поставлених цілей.

## 2. Призначення та умови застосування

Дослідження та моделювання процедур менеджменту програмної інженерії при управлінні проектами має на меті вдосконалення методів керування розробкою програмного забезпечення для досягнення оптимальних результатів. Основні завдання включають аналіз, вивчення та оптимізацію стратегій управління, розробку моделей для прогнозування результатів та забезпечення ефективної координації всіх етапів життєвого циклу проекту.

Умови застосування:

Складність проектів програмної інженерії: Чим більша складність проекту, тим важливіше використання досліджень та моделювання для розробки оптимальних стратегій управління. Складні проекти часто включають велику кількість завдань, ресурсів та ризиків, що вимагає системного підходу до менеджменту.

Забезпечення якості: Дослідження та моделювання дозволяють розробити ефективні методи контролю за якістю продукту. Це важливо для забезпечення високої якості програмного забезпечення та вчасної поставки.

Ризик-менеджмент: Дослідження ризиків та їх моделювання дозволяє підготувати стратегії для зменшення впливу можливих проблем, що може стати вирішальним у виробничих умовах невизначеності.

Оптимізація використання ресурсів: Ефективне управління ресурсами, такими як час, бюджет та людські ресурси, є важливим фактором успішного завершення проекту. Моделювання дозволяє оптимізувати використання цих ресурсів.

Стратегічне планування: Дослідження та моделювання процедур менеджменту допомагає розробляти стратегічні плани для досягнення поставлених цілей, що є важливим у сучасному конкурентному середовищі.

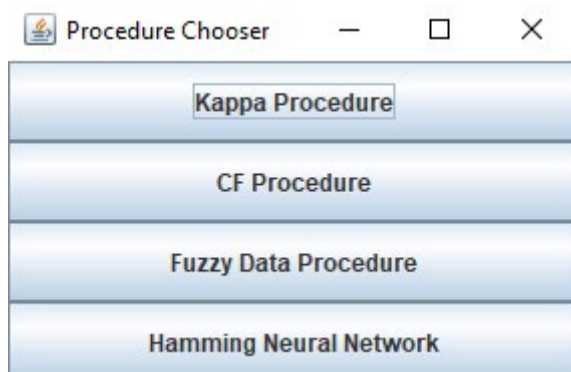
### 3. Підготовка до роботи

Для запуску додатку користувачу необхідно мати Файли Chooser.jar, КарраCalculator.jar, CFProcedure.jar, FuzzyProcedure.jar.

Також необхідне встановлення JDK, Maven та IntelliJ Idea або Docker з необхідним Docker Image.

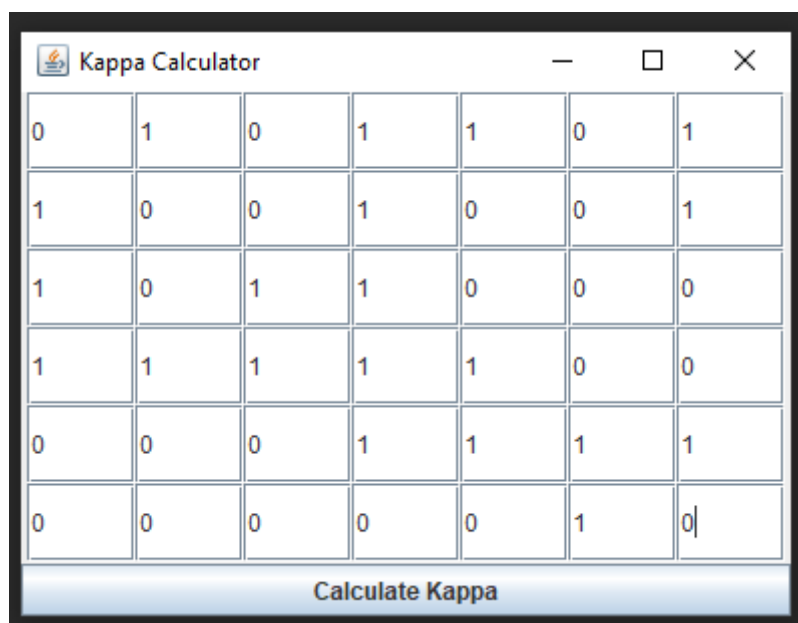
### 4. Опис операцій

Програма має декілька процедур.

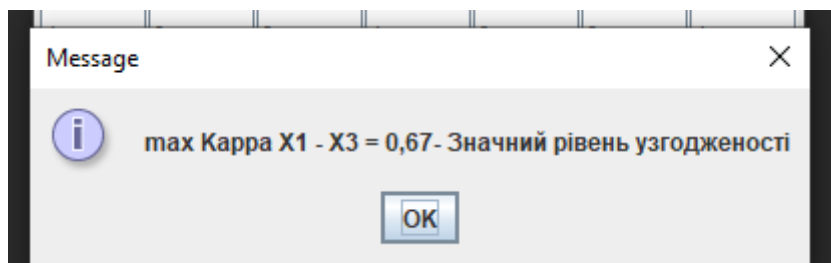


Процедура Каппи Коена

Заповнення даних для проведення проуедури



Після натиснення кнопки розрахунку з'являється вікно з результатом розрахунку



```

X1 - X2 Карра = 0,00
X1 - X3 Карра = 0,67
X1 - X4 Карра = 0,33
X1 - X5 Карра = -0,33
X1 - X6 Карра = -0,67
X1 - X7 Карра = -0,33
X2 - X3 Карра = 0,25
X2 - X4 Карра = 0,18
X2 - X5 Карра = 0,67
X2 - X6 Карра = -0,50
X2 - X7 Карра = 0,00
X3 - X4 Карра = 0,18
X3 - X5 Карра = 0,00
X3 - X6 Карра = -0,50
X3 - X7 Карра = -0,67
X4 - X5 Карра = 0,33
X4 - X6 Карра = -0,36
X4 - X7 Карра = 0,33
X5 - X6 Карра = 0,00
X5 - X7 Карра = 0,33
X6 - X7 Карра = 0,00

The biggest Карра is:
X1 - X3

```

Результат логуювання

Процедура CF

The screenshot shows a window titled "CF Procedure" with the following content:

**Enter vector values:**

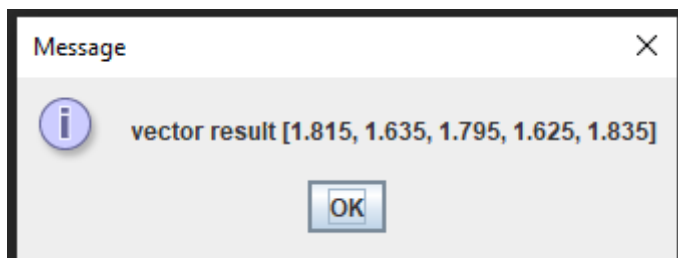
0.00	0.09	0.40	0.17	0.52	0.97
------	------	------	------	------	------

**Enter matrix1 values:**

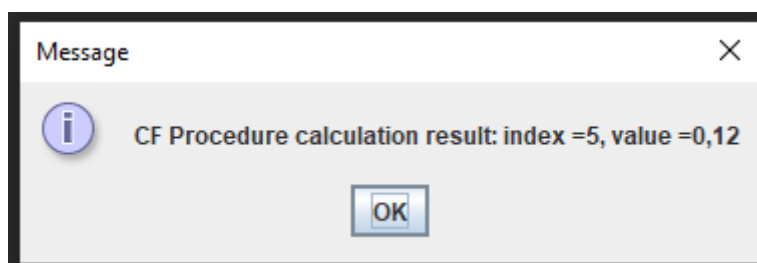
0.69	0.06	0.08	0.51	0.00	0.72
0.17	0.43	0.09	0.81	0.16	0.87
0.33	0.49	0.21	0.90	0.17	0.88
0.05	0.88	0.75	0.44	0.16	0.44
0.50	0.68	0.00	0.84	0.81	0.53

Calculate CF

Після натиснення кнопки розрахунку, виводить результат розрахунку стосовно вектора.



В іншому вікні відкривається результат розрахунку і знаходження відповідності за індексом відповідно до вхідних даних



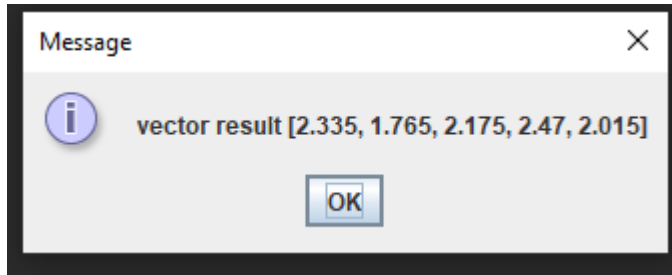
### Процедура Fuzzy

A screenshot of the 'Fuzzy Procedure' application window. The title bar shows the application name and standard window controls. The main area contains two input sections: 'Enter vector values:' with six input fields containing the values 0, 1, 0.3, 0.5, 0.5, and 0.03; and 'Enter matrix1 values:' with a 5x6 grid of input fields containing the following values:

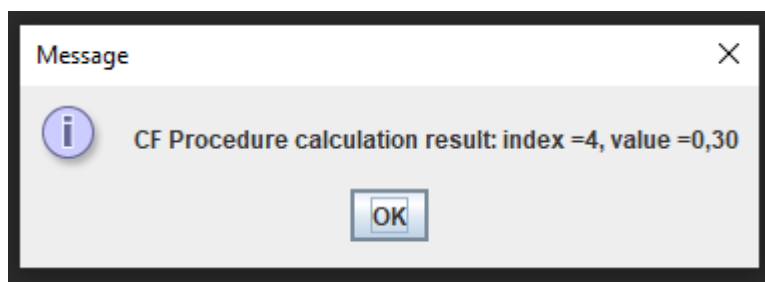
0.15	0.72	0.34	0.38	0.89	0.89
0.21	0.28	0.28	0.89	0.72	0.99
0.70	0.63	0.89	0.72	0.39	0.63
0.75	0.21	0.75	0.63	0.28	0.72
0.89	0.75	0.21	0.99	0.39	0.99

At the bottom center of the window, there is a 'Calculate Fuzzy' button.

Після натиснення кнопки розрахунку, виводить результат розрахунку стосовно вектора.



В іншому вікні відкривається результат розрахунку і знаходження відповідності за індексом відповідно до вхідних даних



## 5. Аварійні ситуації

Якщо програмний засіб буде запускатися з пристрою який не відповідає необхідним технічним характеристикам, буде виведено відповідне повідомлення користувачеві.

Якщо користувач введе невідповідні данні , буде виведено відповідне повідомлення користувачеві.

Якщо користувач закриє програмний засіб, то при наступному запуску програми, його попередній результат не збережеться.

Якщо програмний засіб під час роботи буде поводити некоректно чи із помилкою, від користувача потребується перезапустити додаток для відновлення коректної роботи програмного засобу.

**ДОДАТОК В****ЗАТВЕРДЖУЮ****Проректор****Українського державного  
університету науки і  
технології****Анатолій****РАДКЕВИЧ****ДОСЛІДЖЕННЯ І МОДЕЛЮВАННЯ ПРОЦЕДУР МЕНЕДЖМЕНТУ  
ПРОГРАМНОЇ ІНЖЕНЕРІЇ ПРИ УПРАВЛІННІ ПРОЄКТАМИ****Текст програми****44165850.01345-01 12 01****Завідувач кафедри КІТ****\_\_\_\_\_Вадим ГОРЯЧКІН****Керівник розробки****\_\_\_\_\_Владислав СКАЛОЗУБ****Виконавець****\_\_\_\_\_Ілля ДУДНИК****Нормоконтролер****\_\_\_\_\_Світлана ВОЛКОВА****2024**

Код програми для Дослідження і моделювання процедур  
менеджменту програмної інженерії при управлінні проектами

```

package Final_pack;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Chooser extends JFrame {
    private JButton kappaButton;
    private JButton cfButton;
    private JButton fuzzyButton;

    public Chooser() {
        setTitle("Procedure Chooser");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 150);

        // Set layout manager
        setLayout(new GridLayout(3, 1));

        // Create buttons for each procedure
        kappaButton = new JButton("Kappa Procedure");
        kappaButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                openKappaProcedure();
            }
        });

        cfButton = new JButton("CF Procedure");
        cfButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                openCFProcedure();
            }
        });

        fuzzyButton = new JButton("Fuzzy Data Procedure");
        fuzzyButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                openFuzzyProcedure();
            }
        });

        // Add buttons to the frame
        add(kappaButton);
        add(cfButton);
        add(fuzzyButton);
    }

    private void openKappaProcedure() {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new KappaCalculatorGUI().setVisible(true);
            }
        });
    }
}

```

```

    });
}

private void openCFProcedure() {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new CFProcedureGUI().setVisible(true);
        }
    });
}

private void openFuzzyProcedure() {
    // Implement the code to open the Fuzzy Data procedure GUI here
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new FuzzyProcedureGUI().setVisible(true);
        }
    });
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Chooser().setVisible(true);
        }
    });
}
}

package Final_pack;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Random;

public class KappaCalculatorGUI extends JFrame {
    private JTextField[][] textFieldArray;
    private JButton calculateButton;
    private static String maxKappa;

    public KappaCalculatorGUI() {

        setTitle("Kappa Calculator");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);

        // Set layout manager
        setLayout(new BorderLayout());

        // Create a panel for input table
        JPanel inputPanel = new JPanel();
        inputPanel.setLayout(new GridLayout(6, 7));

```

```

textFieldArray = new JTextField[6][7];

Random random = new Random();

for (int i = 0; i < 6; i++) {
    for (int j = 0; j < 7; j++) {
        textFieldArray[i][j] = new JTextField();
        // Инициализация случайными значениями 0 или 1
        textFieldArray[i][j].setText(random.nextInt(2) + "");
        inputPanel.add(textFieldArray[i][j]);
    }
}

// Create a button to calculate Kappa
calculateButton = new JButton("Calculate Kappa");
calculateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        calculateKappa();
    }
});

// Add components to the frame
add(inputPanel, BorderLayout.CENTER);
add(calculateButton, BorderLayout.SOUTH);
}

private void calculateKappa() {
    int[][] table = new int[6][7];

    // Retrieve values from text fields and populate the table
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < 7; j++) {
            try {
                table[i][j] =
Integer.parseInt(textFieldArray[i][j].getText());
            } catch (NumberFormatException e) {
                // Handle invalid input (non-numeric)
                JOptionPane.showMessageDialog(this, "Invalid input. Please
enter numeric values.");
                return;
            }
        }
    }

    // Call your existing method to calculate Kappa
    double kappa = countTheBiggestKappa(table);
    String s;
    if (kappa < 0.2) {
        s = "Незначний рівень узгодженості";
    } else if (kappa < 0.4) {
        s = "Справедливий рівень узгодженості";
    } else if (kappa < 0.6) {
        s = "Помірний рівень узгодженості";
    } else if (kappa < 0.8) {
        s = "Значний рівень узгодженості";
    } else {
        s = "Близько до ідеального рівня узгодженості";
    }

    // Display the result
    String formattedKappa = String.format("%.2f", kappa) + "";
}

```

```

        JOptionPane.showMessageDialog(this, "max Kappa " + maxKappa + " = " +
formattedKappa + "- " + s);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new KappaCalculatorGUI().setVisible(true);
            }
        });
    }
    private static double countTheBiggestKappa(int[][] table) {
        return countPlusMinus(table);
    }

    private static double countPlusMinus(int[][] table) {

        double result = -110.0;

        int numCols = table[0].length;
        int numRows = table.length;
        int[] matches = new int[4];
        double kappa;

        for (int col = 0; col < numCols; col++) {
            for (int nextCol = col + 1; nextCol < numCols; nextCol++) {
                for (int row = 0; row < numRows; row++) {
                    if (table[row][col] == 1 && table[row][nextCol] == 1) {
                        matches[0]++;
                    } else if (table[row][col] == 1 && table[row][nextCol] == 0)
{
                        matches[1]++;
                    } else if (table[row][col] == 0 && table[row][nextCol] == 1)
{
                        matches[2]++;
                    } else if (table[row][col] == 0 && table[row][nextCol] == 0)
{
                        matches[3]++;
                    }
                }
                kappa = countKappa(matches);
                System.out.print("X" + (col + 1) + " - X" + (nextCol + 1));
                String formatted = String.format("%.2f", kappa);
                System.out.println(" Kappa = " + formatted);

                if (kappa > result) {
                    result = kappa;
                    maxKappa = "X" + (col + 1) + " - X" + (nextCol + 1);
                }
                matches = new int[]{0, 0, 0, 0};
            }
        }
        System.out.println("\nThe biggest Kappa is: ");
        System.out.println(maxKappa);
        return result;
    }

    private static double countKappa(int[] table) {
        double yy = table[0];
        double yn = table[1];
    }

```

```

double ny = table[2];
double nn = table[3];

double countOfAll = (yy + yn + ny + nn);

double p0 = (yy + nn) / countOfAll;

double pYes = ((yy + yn) / countOfAll) * ((yy + ny) / countOfAll);
double pNo = ((nn + yn) / countOfAll) * ((nn + ny) / countOfAll);

double pE = pNo + pYes;

double kappa = (p0 - pE) / (1 - pE);

return kappa;
}
}

```

```
package Final_pack;
```

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Arrays;
import java.util.Random;

```

```

public class CFProcedureGUI extends JFrame {
    private JTextField[] vectorTextFields;
    private JTextField[][] matrix1TextFields;
    private JButton calculateButton;

    public CFProcedureGUI() {
        setTitle("CF Procedure");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setSize(400, 300);

        // Set layout manager
        setLayout(new GridBagLayout());

        // Create label for vector input
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.WEST;
        add(new JLabel("Enter vector values:"), gbc);

        // Create text fields for vector
        vectorTextFields = new JTextField[6];
        JPanel vectorPanel = new JPanel(new GridLayout(1, 6));
        int cellHeight = 25; // Set the desired height for cells (adjust as
needed)
        int cellWidth = 50; // Set the desired width for cells (adjust as
needed)
        for (int i = 0; i < 6; i++) {
            vectorTextFields[i] = new JTextField();
            vectorTextFields[i].setPreferredSize(new Dimension(cellWidth,
cellHeight));
            vectorTextFields[i].setText(String.format("%.2f", new

```

```

Random().nextDouble().replace(",", ".");
    vectorPanel.add(vectorTextFields[i]);
}
gbc.gridx = 0;
gbc.gridy = 1;
gbc.anchor = GridBagConstraints.WEST;
add(vectorPanel, gbc);

// Create label for matrix input
gbc.gridx = 0;
gbc.gridy = 2;
gbc.anchor = GridBagConstraints.WEST;
add(new JLabel("Enter matrix1 values:"), gbc);

// Create text fields for matrix1
matrix1TextFields = new JTextField[5][6];
JPanel matrixPanel = new JPanel(new GridLayout(5, 6));
for (int i = 0; i < 5; i++) {
    for (int j = 0; j < 6; j++) {
        matrix1TextFields[i][j] = new JTextField();
        matrix1TextFields[i][j].setColumns(3); // Set the desired width
        (adjust as needed)
        matrix1TextFields[i][j].setPreferredSize(new
Dimension(matrix1TextFields[i][j].getPreferredSize().width, cellHeight));
        matrix1TextFields[i][j].setText(String.format("%.2f", new
Random().nextDouble().replace(",", ".")));
        matrixPanel.add(matrix1TextFields[i][j]);
    }
}
gbc.gridx = 0;
gbc.gridy = 3;
gbc.anchor = GridBagConstraints.WEST;
add(matrixPanel, gbc);

// Create button to calculate CF
calculateButton = new JButton("Calculate CF");
calculateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        calculateCF();
    }
});
gbc.gridx = 0;
gbc.gridy = 4;
gbc.anchor = GridBagConstraints.CENTER;
add(calculateButton, gbc);
}

private void calculateCF() {
    try {
        // Parse vector and matrix1 from text fields
        double[] vector = new double[6];
        for (int i = 0; i < 6; i++) {
            vector[i] =
Double.parseDouble(vectorTextFields[i].getText().trim());
        }

        double[][] matrix1 = new double[5][6];
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 6; j++) {
                matrix1[i][j] =

```

```

Double.parseDouble(matrix1TextFields[i][j].getText().trim());
        }
    }

    // Call your existing CF procedure code with the parsed vector and
matrix1

    double res = performCF(vector, matrix1);
    String result = String.format("%.2f", res);
    int indexOfResult = performCFIndex(vector, matrix1) + 1;
    double[] vectorRes = performCFVector(vector, matrix1);

    JOptionPane.showMessageDialog(this, "vector result " +
Arrays.toString(vectorRes));

    JOptionPane.showMessageDialog(this, "CF Procedure calculation
result: index =" + indexOfResult + ", value =" + result);

    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Invalid input. Please enter
numeric values.");
    }
}

private double[] performCFVector(double[] vector, double[][] matrix1) {
    double[][] matrix2 = {
        {1, -0.25, -0.25, -0.25, -0.25},
        {-0.25, 1, -0.25, -0.25, -0.25},
        {-0.25, -0.25, 1, -0.25, -0.25},
        {-0.25, -0.25, -0.25, 1, -0.25},
        {-0.25, -0.25, -0.25, -0.25, 1}
    };
    double t = 0.0;
    double[] firstResult = calculateFrist(vector, matrix1, t);
    return firstResult;
}

private int performCFIndex(double[] vector, double[][] matrix1) {
    double[][] matrix2 = {
        {1, -0.25, -0.25, -0.25, -0.25},
        {-0.25, 1, -0.25, -0.25, -0.25},
        {-0.25, -0.25, 1, -0.25, -0.25},
        {-0.25, -0.25, -0.25, 1, -0.25},
        {-0.25, -0.25, -0.25, -0.25, 1}
    };
    double t = 0.0;

    double[] firstResult = calculateFrist(vector, matrix1, t);

    System.out.println("Output of the first layer:");
    for (int j = 0; j < firstResult.length; j++) {
        System.out.println("Neuron " + (j + 1) + ": " + firstResult[j]);
    }

    System.out.println();

    double[] indexOfResult = findResultIndex(matrix2, firstResult);

    for (int i = 0; i < indexOfResult.length; i++) {
        if (indexOfResult[i] != 0) {
            return i;
        }
    }
}

```

```

    }
}

return -100;
}

private double performCF(double[] vector, double[][] matrix1) {
    double[][] matrix2 = {
        {1, -0.25, -0.25, -0.25, -0.25},
        {-0.25, 1, -0.25, -0.25, -0.25},
        {-0.25, -0.25, 1, -0.25, -0.25},
        {-0.25, -0.25, -0.25, 1, -0.25},
        {-0.25, -0.25, -0.25, -0.25, 1}
    };
    double t = 0.0;

    double[] firstResult = calculateFrist(vector, matrix1, t);

    System.out.println("Output of the first layer:");
    for (int j = 0; j < firstResult.length; j++) {
        System.out.println("Neuron " + (j + 1) + ": " + firstResult[j]);
    }

    System.out.println();

    double[] indexOfResult = findResultIndex(matrix2, firstResult);

    for (int i = 0; i < indexOfResult.length; i++) {
        if (indexOfResult[i] != 0) {
            return indexOfResult[i];
        }
    }

    return -100;
}

static double[] findResultIndex(double[][] matrix2, double[] firstResult) {
    double[] result = new double[firstResult.length];
    for (int i = 0; i < result.length; i++) {
        if (firstResult[i] <= 0) {
            result[i] = 0.0;
        } else {
            result[i] = firstResult[i];
        }
    }
    int columns = firstResult.length;

    int rows = matrix2.length;

    int counter = 0;

    double[] lastLoop = new double[result.length];

    double[] res = new double[result.length];

    while (counter > 1 || counter == 0) {
        counter = 0;
        System.arraycopy(result, 0, lastLoop, 0, result.length);
        System.arraycopy(result, 0, res, 0, result.length);
    }
}

```

```

        for (int j = 0; j < rows; j++) {
            double count = 0;
            for (int i = 0; i < columns; i++) {
                count += matrix2[j][i] * res[i];
            }

            if (count <= 0) {
                result[j] = 0.0;
            } else {
                result[j] = count;
                counter++;
            }

            if (findMaxIndexes(result).length == 1){
                return findMaxIndexesWithoutTrimm(result);
            }
            if (counter == 0) {
                return lastLoop;
            }
        }

        return result;
    }

    private static double[] findMaxIndexes(double[] indexOfResult) {

        double[] arr = new double[indexOfResult.length];

        double max = 0;

        int count =0;

        for (int i = 0; i < arr.length; i++) {
            if (max < indexOfResult[i]){
                max = indexOfResult[i];
            }
        }

        for (int i = 0; i < arr.length; i++) {
            if (max - indexOfResult[i] < max * 0.1 || indexOfResult[i] >= 0.1){
                arr[i] = indexOfResult[i];
                count++;
            }
            else{
                arr[i] = 0.0;
            }
        }

        double[] result = new double[count];

        for (int i = 0, a = 0; i < arr.length ; i++) {
            if (arr[i] > 0){
                result[a] = arr[i];
            }
        }

        return result;
    }

    private static double[] findMaxIndexesWithoutTrimm(double[] indexOfResult) {

```

```

double[] arr = new double[indexOfResult.length];

double max = 0;

int count =0;

for (int i = 0; i < arr.length; i++) {
    if (max < indexOfResult[i]){
        max = indexOfResult[i];
    }
}
for (int i = 0; i < arr.length; i++) {
    if (max - indexOfResult[i] < max * 0.1 || indexOfResult[i] >= 0.1){
        arr[i] = indexOfResult[i];
        count++;
    }
    else{
        arr[i] = 0.0;
    }
}

return arr;
}
static double[] calculateFrist(double[] vector, double[][] matrix1, double
t) {
    // довжина вектору
    int columns = vector.length;
    // висота матриці
    int rows = matrix1.length;

    // результуючий вектор
    double[] result = new double[rows];

    // Цикл розрахунку за формулою
    for (int j = 0; j < rows; j++) {
        double sum = 0.0;

        for (int i = 0; i < columns; i++) {
            sum += (1 - Math.abs(matrix1[j][i] - vector[i]));
        }

        result[j] = (sum + t) / 2;
    }

    return result;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new CFProcedureGUI().setVisible(true);
        }
    });
}

}

package Final_pack;
import javax.swing.*.*;
import java.awt.*.*;

```

```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Arrays;

public class FuzzyProcedureGUI extends JFrame {
    private JTextField[] vectorTextFields;
    private JTextField[][] matrix1TextFields;
    private JButton calculateButton;

    public FuzzyProcedureGUI() {
        setTitle("Fuzzy Procedure");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setSize(400, 300);

        // Set layout manager
        setLayout(new GridBagLayout());

        // Create label for vector input
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.gridx = 0;
        gbc.gridy = 0;
        gbc.anchor = GridBagConstraints.WEST;
        add(new JLabel("Enter vector values:"), gbc);

        // Create text fields for vector
        vectorTextFields = new JTextField[6];
        JPanel vectorPanel = new JPanel(new GridLayout(1, 6));
        int cellHeight = 25; // Set the desired height for cells (adjust as
needed)
        int cellWidth = 50; // Set the desired width for cells (adjust as
needed)
        for (int i = 0; i < 6; i++) {
            vectorTextFields[i] = new JTextField();
            vectorTextFields[i].setPreferredSize(new Dimension(cellWidth,
cellHeight));
            vectorPanel.add(vectorTextFields[i]);
        }
        gbc.gridx = 0;
        gbc.gridy = 1;
        gbc.anchor = GridBagConstraints.WEST;
        add(vectorPanel, gbc);

        // Create label for matrix input
        gbc.gridx = 0;
        gbc.gridy = 2;
        gbc.anchor = GridBagConstraints.WEST;
        add(new JLabel("Enter matrix1 values:"), gbc);

        // Create text fields for matrix1
        matrix1TextFields = new JTextField[5][6];
        JPanel matrixPanel = new JPanel(new GridLayout(5, 6));
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 6; j++) {
                matrix1TextFields[i][j] = new JTextField();
                matrix1TextFields[i][j].setColumns(3); // Set the desired width
(adjust as needed)
                matrix1TextFields[i][j].setPreferredSize(new
Dimension(matrix1TextFields[i][j].getPreferredSize().width, cellHeight));
                matrixPanel.add(matrix1TextFields[i][j]);
            }
        }
    }
}

```

```

gbc.gridx = 0;
gbc.gridy = 3;
gbc.anchor = GridBagConstraints.WEST;
add(matrixPanel, gbc);

// Create button to calculate CF
calculateButton = new JButton("Calculate Fuzzy");
calculateButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        calculateCF();
    }
});
gbc.gridx = 0;
gbc.gridy = 4;
gbc.anchor = GridBagConstraints.CENTER;
add(calculateButton, gbc);
}

private void calculateCF() {
    try {
        // Parse vector and matrix1 from text fields
        double[] vector = new double[6];
        for (int i = 0; i < 6; i++) {
            vector[i] =
Double.parseDouble(vectorTextFields[i].getText().trim());
        }

        double[][] matrix1 = new double[5][6];
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 6; j++) {
                matrix1[i][j] =
Double.parseDouble(matrix1TextFields[i][j].getText().trim());
            }
        }

        // Call your existing CF procedure code with the parsed vector and
matrix1

        double res = performCF(vector, matrix1);
        String result = String.format("%.2f", res);
        int indexOfResult = performCFIndex(vector, matrix1) + 1;
        double[] vectorRes = performCFVector(vector, matrix1);

        JOptionPane.showMessageDialog(this, "vector result " +
Arrays.toString(vectorRes));

        JOptionPane.showMessageDialog(this, "Fuzzy Procedure calculation
result: index =" + indexOfResult + ", value =" + result);

    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Invalid input. Please enter
numeric values.");
    }
}

private double[] performCFVector(double[] vector, double[][] matrix1) {
    double[][] matrix2 = {
        {1, -0.25, -0.25, -0.25, -0.25},
        {-0.25, 1, -0.25, -0.25, -0.25},
        {-0.25, -0.25, 1, -0.25, -0.25},
    }
}

```

```

        {-0.25, -0.25, -0.25, 1, -0.25,},
        {-0.25, -0.25, -0.25, -0.25, 1}
    };
    double t = 0.0;

    double[] firstResult = calculateFrist(vector, matrix1, t);
    return firstResult;
}

private int performCFIndex(double[] vector, double[][] matrix1) {
    double[][] matrix2 = {
        {1, -0.25, -0.25, -0.25, -0.25},
        {-0.25, 1, -0.25, -0.25, -0.25},
        {-0.25, -0.25, 1, -0.25, -0.25},
        {-0.25, -0.25, -0.25, 1, -0.25,},
        {-0.25, -0.25, -0.25, -0.25, 1}
    };
    double t = 0.0;

    double[] firstResult = calculateFrist(vector, matrix1, t);

    System.out.println("Output of the first layer:");
    for (int j = 0; j < firstResult.length; j++) {
        System.out.println("Neuron " + (j + 1) + ": " + firstResult[j]);
    }

    System.out.println();

    double[] indexOfResult = findResultIndex(matrix2, firstResult);

    for (int i = 0; i < indexOfResult.length; i++) {
        if (indexOfResult[i] != 0) {
            return i;
        }
    }

    return -100;
}

private double performCF(double[] vector, double[][] matrix1) {
    double[][] matrix2 = {
        {1, -0.25, -0.25, -0.25, -0.25},
        {-0.25, 1, -0.25, -0.25, -0.25},
        {-0.25, -0.25, 1, -0.25, -0.25},
        {-0.25, -0.25, -0.25, 1, -0.25,},
        {-0.25, -0.25, -0.25, -0.25, 1}
    };
    double t = 0.0;

    double[] firstResult = calculateFrist(vector, matrix1, t);

    System.out.println("Output of the first layer:");
    for (int j = 0; j < firstResult.length; j++) {
        System.out.println("Neuron " + (j + 1) + ": " + firstResult[j]);
    }

    System.out.println();

    double[] indexOfResult = findResultIndex(matrix2, firstResult);

```

```

    for (int i = 0; i < indexOfResult.length; i++) {
        if (indexOfResult[i] != 0) {
            return indexOfResult[i];
        }
    }

    return -100;
}

static double[] findResultIndex(double[][] matrix2, double[] firstResult) {
    double[] result = new double[firstResult.length];
    for (int i = 0; i < result.length; i++) {
        if (firstResult[i] <= 0) {
            result[i] = 0.0;
        } else {
            result[i] = firstResult[i];
        }
    }
    int columns = firstResult.length;

    int rows = matrix2.length;

    int counter = 0;

    double[] lastLoop = new double[result.length];

    double[] res = new double[result.length];

    while (counter > 1 || counter == 0) {
        counter = 0;
        System.arraycopy(result, 0, lastLoop, 0, result.length);
        System.arraycopy(result, 0, res, 0, result.length);

        for (int j = 0; j < rows; j++) {
            double count = 0;
            for (int i = 0; i < columns; i++) {
                count += matrix2[j][i] * res[i];
            }

            if (count <= 0) {
                result[j] = 0.0;
            } else {
                result[j] = count;
                counter++;
            }
        }

        if (findMaxIndexes(result).length == 1) {
            return findMaxIndexesWithoutTrimm(result);
        }

        if (counter == 0) {
            return lastLoop;
        }
    }

    return result;
}

```

```

}

private static double[] findMaxIndexes(double[] indexOfResult) {

    double[] arr = new double[indexOfResult.length];

    double max = 0;

    int count =0;

    for (int i = 0; i < arr.length; i++) {
        if (max < indexOfResult[i]){
            max = indexOfResult[i];
        }
    }
    for (int i = 0; i < arr.length; i++) {
        if (max - indexOfResult[i] < max * 0.1 || indexOfResult[i] >= 0.1){
            arr[i] = indexOfResult[i];
            count++;
        }
        else{
            arr[i] = 0.0;
        }
    }

    double[] result = new double[count];

    for (int i = 0, a = 0; i < arr.length ; i++) {
        if (arr[i] > 0){
            result[a] = arr[i];
        }
    }

    return result;
}

private static double[] findMaxIndexesWithoutTrimm(double[] indexOfResult) {

    double[] arr = new double[indexOfResult.length];

    double max = 0;

    int count =0;

    for (int i = 0; i < arr.length; i++) {
        if (max < indexOfResult[i]){
            max = indexOfResult[i];
        }
    }
    for (int i = 0; i < arr.length; i++) {
        if (max - indexOfResult[i] < max * 0.1 || indexOfResult[i] >= 0.1){
            arr[i] = indexOfResult[i];
            count++;
        }
        else{
            arr[i] = 0.0;
        }
    }

    return arr;
}

static double[] calculateFrist(double[] vector, double[][] matrix1, double

```

```

t) {
    // довжина вектору
    int columns = vector.length;
    // висота матриці
    int rows = matrix1.length;

    // результуючий вектор
    double[] result = new double[rows];

    // Цикл розрахунку за формулою
    for (int j = 0; j < rows; j++) {
        double sum = 0.0;

        for (int i = 0; i < columns; i++) {
            sum += (1 - Math.abs(matrix1[j][i] - vector[i]));
        }

        result[j] = (sum + t) / 2;
    }

    return result;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new CFProcedureGUI().setVisible(true);
        }
    });
}
}

```

## ДОДАТОК Г

Результати апробації результатів досліджень



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
МІНІСТЕРСТВО ІНФРАСТРУКТУРИ УКРАЇНИ  
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ ТА ТЕХНОЛОГІЙ  
СХІДНИЙ НАУКОВИЙ ЦЕНТР ТРАНСПОРТНОЇ АКАДЕМІЇ НАУК



### ЗМІСТ

Дослідження можливості мурашиного методу для визначення топології бездротової локальної мережі на сортувальній станції **Помилка! Закладку не визначено.**

Пахомова В. М., Салогуб М. В., Український державний університет науки і технологій

Прогнозування затримки на маршрутизаторі в інформаційно-телекомунікаційній системі залізничного транспорту засобами нейронечіткої мережі **Помилка! Закладку не визначено.**

Пахомова В. М., Хрестян А. В., Український державний університет науки і технологій

Розробка програмного забезпечення для прогнозування користувацьких дій у виробничому середовищі ..... **Помилка! Закладку не визначено.**

Погребняк В. М., Горячкін В. М., Український державний університет науки і технологій

Що привносить когнітивність до основ нечіткої математики? **Помилка! Закладку не визначено.**

Прокопчук Ю. О., Інститут технічної механіки НАН і ДКА України, Україна

Дослідження методів та засобів аутентифікації користувача у реальному часі шляхом розпізнавання обличчя ..... **Помилка! Закладку не визначено.**

Різниченко М. О., Горбова О. В., Український державний університет науки і технологій, Україна

Поведінкові стратегії Інтернет бізнесу ..... **Помилка! Закладку не визначено.**

Савчук Л. М., Ковальчук Є. В. Український державний університет науки і технологій, Інститут промислових та бізнес технологій, Україна

Вибір цілі, критерію і норми при проектуванні інтелектуальної системи управління організаційно-технічними процесами ..... **Помилка! Закладку не визначено.**

Самойлов С.П., Український державний університет науки і технологій, Україна

Розпізнавання рукописних символів за допомогою нейронної мережі **Помилка! Закладку не визначено.**

Середа О.А., Горячкін В.М., Український державний університет науки і технологій

Підвищення якості коду мобільних застосунків методами рефакторингу з використанням генеративного штучного інтелекту ..... **Помилка! Закладку не визначено.**

Сирота О. А., Горячкін В.М., Український державний університет науки і технологій

Багатовимірні нечіткі моделі моніторингу і реабілітації з нерівномірним інтервалом спостережень..... **Помилка! Закладку не визначено.**

Скалозуб В.В., Горячкін В.М., Клименко І.В., Мурашов О.В., Український державний університет науки і технологій

Процедури редукції і каппа статистики у завданнях формування достовірних математичних моделей класифікації невизначених даних**Помилка! Закладку не визначено.**

Скалозуб В.В., Терлецький І.А., Дудник І.О., Український державний університет науки і технологій, Скалозуб М. PayPal, Швеція

**Процедури редукції і каппа статистики у завданнях формування достовірних математичних моделей класифікації невизначених даних**

Скалозуб В.В., Терлецький І.А., Дудник І.О.,

Український державний університет науки і технологій,

Скалозуб М. PayPal, Швеція

Завдання та процедури класифікації та діагностування за умов неповної визначеності вихідних даних є досить поширеними на практиці. За їх результатами формуються моделі оптимального керування різноманітними технологічними процесами, регулювання у системах обслуговування, вибору раціональних заходів/виконавців тощо. Разом з тим одним із актуальних нових аспектів моделей діагностування являється узгодженість ймовірнісних вимог до достовірності результатів, розмірності параметрів шаблонів, а також кількості даних, що використовуються для розрахунків класів-переможців. Для реалізації моделей класифікації з невизначеними даними в роботі застосовуються удосконалені моделі асоціативної пам'яті Хеммінга, МХН. У МХН використовуються в якості моделей даних нечіткі множини ( $\mu_X: X \rightarrow [0; 1]$ ), а також коефіцієнти впевненості  $CF(A)$  із значеннями в множині  $[-1; +1]$ . При тому класичні моделі Хеммінга представляють дані за допомогою дискретної множини  $\{-1; +1\}$ .

У представлених нами удосконалених моделях МХН реалізоване також завдання із обґрунтування структури шаблонів багатопараметричної класифікації. У доповіді приведені удосконалені математичні моделі, алгоритми та програмні засоби, призначені для підвищення достовірності результатів класифікації при невизначених даних у формі розмитих/нечітких величин та коефіцієнтів упевненості  $CF(A)$ . Метою удосконалення являється потреба урахувати граничні розмірності та склад параметрів моделі класифікації таким чином, щоб забезпечити встановлені ймовірнісні вимоги щодо достовірності результатів розрахунків. Зазначені властивості моделей забезпечується шляхом модифікації нейронних мереж Хеммінга, а також застосування процедур редукції розмірності простору

параметрів моделей і статистики каппа Коена, які дозволяють сформувати моделі завдань класифікації із заданими ймовірнісними вимогами.

При тому були розроблені нові постановки, математичні моделі, а також виконано класифікації за нечіткими даними, що вирішують завдання із встановлення авторів україномовних текстів і завдань вибору кандидата за моделлю даних у форматі коефіцієнтів  $CF(A)$ . В якості моделі класифікації за нечіткими даними розглядається завдання щодо визначення авторства творів/текстів україномовних авторів (ЗАТ). Особливістю моделей ЗАТ являється відсутність одного загального шаблону авторів (ураховані множини шаблонів/зразків для «авторів»), відсутні і вимоги до числа етапів процедури, за якою реалізується завдання визначення автора твору. Раніше завдання вирішувалося за один крок, тож була неможливою процедура редукції.

Процедура редукції у цілому складається із таких етапів: - 1) виконати оцінку показника «каппа» ступеню подібності результатів класифікації на основі моделей шаблонів з різним числом або складом параметрів, - 2) при забезпеченні зі встановленим ступенем достовірності «подібності» результатів класифікації для різних моделей-шаблонів можливо виконати скорочення моделі, залишити один із шаблонів, - 3) визначити і видалити найменше значимі або найбільше «подібні» між собою параметри моделі класифікації, враховуючи граничне значення розмірності простору моделей « $p_0$ ».

Розроблено програмні засоби для формування наведених моделей класифікації при невизначених даних процедурами редукції і каппа статистики. Програмний комплекс призначений для побудови математичних моделей процесів класифікації – шаблонів баз знань для реалізації класифікації, розрахунків процесів класифікації на основі мережі МХН. Комплекс забезпечує автоматизацію обробки завдань класифікації нечітких та даних формату  $CF(A)$ , виконує класифікацію на основі неточно визначених ознак «еталонів» щодо класів «виконавця» або «автора» україномовного твору.

Системні технології. – Дніпро, УДУНТ, 2023. – № 5 (148). – С. 141-155.

<https://journals.nmetau.edu.ua/index.php/st/article/view/148>

УДК 004.6: 007.5: 004.94

Скалозуб В.В., Горячкін В.М., Терлецький І.А., Дудник І.П.

## **ФОРМУВАННЯ МОДЕЛЕЙ КЛАСИФІКАЦІЇ НЕВИЗНАЧЕНИХ ДАНИХ ПРОЦЕДУРАМИ РЕДУКЦІЇ І КАППА СТАТИСТИКИ**

*Стаття присвячена розвитку математичних моделей класифікації невизначених даних, представлених нечіткими величинами та коефіцієнтами упевненості  $CF(A)$ . Процедури формування шаблонів діагностування використовують модифіковані мережі Хеммінга (МХН), а також методи редукції та статистики каппа Коена. При цьому визначаються граничні розмірності та склад параметрів моделі класифікації, які забезпечують встановлені ймовірнісні вимоги достовірності результатів розрахунків. Представлена процедура редукції простору моделі діагностування невизначених даних. У статті наведено постановки, математичні моделі та реалізації завдань класифікації за недетермінованими даними. Прикладом моделі класифікації за нечіткими даними являється завдання із встановлення авторів україномовних текстів. Завдання класифікації при даних у форматі  $CF(A)$  відповідає відбору кандидата. Результати числового моделювання дозволили встановити результативність, достовірність та ефективність запропонованих процедур формування достовірних моделей класифікації при невизначених даних.*

*Ключові слова: класифікація, достовірні моделі, розмірність простору, нечіткі величини, коефіцієнти  $CF(A)$ , модифікована мережа Хеммінга, процедура редукції, статистика каппа Коена, україномовні тексти, визначення автора, комп'ютерне моделювання.*

## I. Вступ та постановка проблеми

Завдання та процедури класифікації та діагностування за умов неповної визначеності вихідних даних (збурені, неповні ін.) являються досить поширеними на практиці [1, 2, 4, 6]. За їх результатами формуються моделі оптимального керування різноманітними технологічними процесами, вибору раціональних заходів/виконавців тощо [1, 6, 9]. Відзначається одна з головних проблем завдань класифікації, яка полягає у встановленні властивостей достовірності та повноти моделей (формати, структура та кількість зразків тощо, які забезпечують достовірний результати). Одним із актуальних нових аспектів моделей діагностування являється узгодженість ймовірнісних вимог до достовірності результатів, розмірності параметрів шаблонів, а також кількості даних (класів) за якими виконується діагностування. Для підвищення достовірності результатів та ефективності класифікації на основі модифікованих мереж Хеммінга (МХН) актуальними являються дослідження можливостей і застосування вимог статистики каппа Коена та методу редукції розмірності [1, 3]. Важливе значення мають постановки завдань керування сервісними системами (С&С) з неповними та неточно визначеними даними і даними в природньомовній формі, які дозволяють представити рішення на основі моделей класифікації.

**Аналіз останніх досліджень і публікацій.** У сучасних складних системах деякі параметри станів або контрольовані характеристики процесів можуть мати значну ступінь невизначеності [1, 5, 6, 9]. Створення інтелектуальних процедур та інформаційної технології (ІТ) для оптимізації потоків замовлень у С&С запропоновано у [1]. В ній вибір керувань виконується шляхом реалізації завдань діагностування з урахуванням умов невизначеності на основі МХН.

Завдання розвитку моделей та інтелектуальних багатопараметричних процедур діагностування за неповними і збуреними даними являються натепер актуальними [1, 2, 6]. У статті [6] було запропоновано підхід до формування шаблонів моделей класифікації на основі МХН [1]. При тому відзначено необхідність застосування методу редукції [3] та статистики каппа Коена [4]. Ці методи разом забезпечують отримання результатів із встановленими гарантіями щодо достовірності результатів. При тому в шаблонах МХН використовуються дані виду коефіцієнтів впевненості,  $CF(A)$  [5].

Удосконалені моделі МХН мають суттєву відмінність від класичних моделей асоціативної пам'яті Хеммінга в завданнях класифікації за неповними та збуреними даними. В МХН використовуються в якості моделей даних нечіткі множини ( $\mu_X: X \rightarrow [0; 1]$ ), а також коефіцієнти впевненості  $CF(A)$  із значеннями в множині  $[-1; +1]$ . При

тому класичні моделі Хеммінга [10] представляють дані за допомогою дискретної множини  $\{-1; +1\}$ . У [1, 6] удосконалено математичні моделі та процедури класифікації МХН з урахуванням показників, які відображають неточність або природномовну структуру первинних даних. Також представлено програмні засоби і результати експериментальних досліджень. За ними встановлена надмірність параметрів-ознак шаблонів, які використовувались для завдань визначення авторства україномовних текстів [7].

Для оцінки граничної розмірності параметрів моделі класифікації « $n_0$ », простору який синтезується, використовуються результати теореми Вапника-Червоненкіса і процедури методу граничних спрощень (МСП) [3]. Величина

$$n_0 = (\varepsilon * L + \ln(h)) / \ln(m) \quad (1)$$

визначає розмірність простору, перевищення якої призводить до втрати гарантії досягнення заданих параметрів достовірності « $\varepsilon, h$ »;  $(1 - h)$  оцінка ймовірності умови безпомилкового розділення випадкової і незалежної вибірки довжини « $L$ » при заданій граничній величині помилкової класифікації « $\varepsilon$ ».

В цій статті розроблена нова процедура, яка забезпечує формування простору (1) багатопараметричної класифікації з указаними вимогам достовірності.

**Мета дослідження** – підвищення достовірності моделей класифікації при невизначених даних, представлених нечіткими величинами та коефіцієнтами упевненості  $CF(A)$ , шляхом застосування процедур редукції і статистики каппа Коена. Отримати постановки, моделі та реалізації завдань класифікації за нечіткими даними, які забезпечують встановлення авторів україномовних текстів, реалізувати завдання вибору кандидата за даними у форматі  $CF(A)$ .

**Результати та основний матеріал дослідження.** Головним завдання, що вирішене у роботі, було формування достовірних математичних моделей класифікації та процедури аналізу при невизначених даних певних типів на основі методів редукції та каппа статистики (ПКР). ПКР складається із таких етапів:

- 1) виконати оцінку показника «каппа» ступеню подібності результатів класифікації на основі моделей шаблонів з різним числом або складом параметрів,
- 2) при забезпеченні «подібності» результатів класифікації для різних моделей-шаблонів виконати скорочення моделі, залишити один із шаблонів,
- 3) визначити і видалити найменше значимі або найбільше «подібні» між собою параметри моделі класифікації, враховуючи значення « $n_0$ ».

Узагальнений варіант циклу скорочення розмірності «шаблонів» моделі при вхідному «еталон»/»вимоги». Нехай  $N(t) > n_0$  число параметрів шаблону на етапі (t) розмірності  $N(t=0) = «m»$ . Утворюється множина конкуруючих моделей шаблонів меншої розмірності. Для цього з набору змінних шаблонів  $N(t)$  і «еталону» вимог

вилучається, наприклад, параметр  $X_j$ ; нові вектори для цих спрощених моделей шаблонів позначимо  $(X/X_j)$ . Такі спрощені моделі формуються для кожної змінної із числа  $N(t)$ , а за ними виконується класифікація на основі МНХ.

Перевірити «подібність» оцінок за каппа статистикою результатів класифікації для усіх пар змінних шаблонів  $\{(X/X_j)\}$ . Для різних наборів змінних  $(X/X_j)$  шаблонів за каппа-оцінками для кожного шаблону призначаються значення «+» або «-» в залежності від результатів класифікації. На основі порівняльного аналізу для всіх змінних утворюють таблицю розбіжностей, за якою розраховують статистичні оцінки «каппа Коена» [4]:

$$K = (P_0 - P_e) / (1 - P_e), \quad (2)$$

В (4)  $P_0$  – ймовірнісна оцінка що показує наскільки спостережувана узгодженість краща за випадкову, а  $P_e$  – результат підрахунку максимально можливої узгодженості за винятком випадкової узгодженості конкуруючих шаблонів [4].

Для вибору параметра спрощення моделі  $N(t)$  визначається пара наборів змінних виду  $(X/X_j)$  із найбільшим значенням «К» (2) та експертна оцінка «подібності» скорочених наборів  $(X/X_j)$ . Якщо величина оцінки пари (2) відповідає вимогам, то певна змінна (їх множина) може бути видалена з «шаблонів» та «еталону». Після видалення таких параметрів з моделі класифікації увесь цикл розрахунків повторюється до виконання умови ( $m \leq n_0$ ).

З метою удосконалення мережі Хеммінга (МХ) при встановленні оцінок величин відстані між нечіткими елементами ( $\mu X$ ) зразків ( $w_{ik}$ ) і вхідним вектором  $X = \{x_i: i=0...n-1\}$  уведено нечітке відношення наступного виду

$$R(W, X) = \{\mu R(w_i, x_i) / \{w_i, x_i\} = (1 - \text{abs}(w_i - x_i)) / \{w_i, x_i\}, i=0, 1, \dots, n-1. \quad (3)$$

Якщо значення ступенів приналежності величин  $\{w_i, x_i\}$  однакові, тоді значення  $\mu R(w_i, x_i) = 1$ . Коли одна з величин  $\{w_i, x_i\}$  дорівнює 0, а інша 1, тоді значення  $\mu R(w_i, x_i) = 0$ , інакше  $\mu R(w_i, x_i) \rightarrow [0; 1]$ . Відношення (3) є одною із можливостей реалізації МХН для формування нечітких моделей класифікації, а також моделей представлених коефіцієнтами упевненості  $CF(A)$  [5].

Таблиця 1

Ki	Li	X	X/{X1}	X/{X2}	X/{X3}	X/{X4}	X/{X5}	X/{X6}
K1	L11		+					+
	L12	+		+		+		
	L13				+			+
K2	L21					+		+
	L22		+	+				
	L23				+			

	L24	+				+		
K3	L31		+	+				
	L32	+					+	
K4	L41							
	L42		+		+	+		
	L43							
	L44	+				+		+
	L45					+		
K5	L51			+				
	L52						+	
	L53							
K6	L61			+				
	L62				+	+		
	L63	+						
	L64				+			
K7	L71						+	
	L72				+			
	L73	+						+
K8	L81					+		
	L82	+	+	+				

### Результати циклу процедури редукції при формуванні моделі класифікації

Для пояснення процедури редукції приведемо умовний приклад реалізації наведених вище завдань на основі табл. 1 результатів класифікації для шаблонів з параметрів  $(X_1, X_2, \dots, X_6)$ . В табл. 1 позначено :  $K_i$  – класи/зразки моделі, які можуть містити кілька екземплярів  $L_i$ ;  $X$  – набір параметрів моделі класифікації на поточному етапі процесу редукції;  $X/\{X_j\}$  – скорочені набори моделей класифікації та вхідних векторів на поточному етапі без множини параметрів  $\{X_j\}$ , які видалені з моделей; знаки «+» – визначення модифікованих шаблонів переможців (класів) при застосуванні моделі МХН. У стовпці  $X$  позначаються шаблони, які були визначені на основі  $X$  наборів параметрів. Саме серед скорочених моделей шаблонів  $X/\{X_j\}$  визначається набір параметрів  $\{X_j\}$ , які необхідно видалити на наступному етапі процедури редукції.

Таблиця 2

Розрахунки показників каппа статистики за таблицями розбіжностей

Розрахунок за класами ( $KM_1, C_2$ )

$$K(X_1/X_2) = 0,143$$

	Так	ні
так	4	1
ні	2	1

Розрахунок за класами ( $KM_1, C_1$ )

$$K(X_1/X_2) = 0,143$$

	Так	ні
так	4	2
ні	1	1

$K(X_1/X_3)$

$K(X_1/X_3) = -0,067$	<table border="1"> <tr><td></td><td>Так</td><td>ні</td></tr> <tr><td>так</td><td>3</td><td>2</td></tr> <tr><td>ні</td><td>2</td><td>1</td></tr> </table>		Так	ні	так	3	2	ні	2	1	$K(X_1/X_3) = -0,333$	<table border="1"> <tr><td></td><td>Так</td><td>ні</td></tr> <tr><td>так</td><td>4</td><td>2</td></tr> <tr><td>ні</td><td>2</td><td>0</td></tr> </table>		Так	ні	так	4	2	ні	2	0
	Так	ні																			
так	3	2																			
ні	2	1																			
	Так	ні																			
так	4	2																			
ні	2	0																			
$K(X_1/X_4)$		$K(X_1/X_4) = 0,333$																			
$K(X_1/X_4) = 0,467$	<table border="1"> <tr><td></td><td>Так</td><td>ні</td></tr> <tr><td>так</td><td>4</td><td>1</td></tr> <tr><td>ні</td><td>1</td><td>2</td></tr> </table>		Так	ні	так	4	1	ні	1	2	$K(X_1/X_4) = 0,333$	<table border="1"> <tr><td></td><td>Так</td><td>ні</td></tr> <tr><td>так</td><td>5</td><td>1</td></tr> <tr><td>ні</td><td>1</td><td>1</td></tr> </table>		Так	ні	так	5	1	ні	1	1
	Так	ні																			
так	4	1																			
ні	1	2																			
	Так	ні																			
так	5	1																			
ні	1	1																			

Множини  $\{X_j\}$  можуть містити не одну, а любую кількість змінних, встановлену у завданні. Прийняте в табл. 1 число параметрів моделі класифікації  $n=6$  – умовне, а граничне значення  $n_0 = 4$ . Дані табл. 1 фіксують результати класифікацій за мережею МХН, які отримані при однакових вихідних постановках завдань для всіх конкуруючих варіантів, сутність яких зараз не суттєва.

Множина варіантів моделей класифікації (КМ) містить всі комбінації пар скорочених моделей  $X/\{X_j\}$ . Виконуються наступні передумови аналізу результатів. КМ<sub>1</sub> – оцінюється класифікація хоча б одним із шаблонів  $L_{ij}$  класу  $K_i$ , КМ<sub>2</sub> – оцінюється класифікація кожним із  $L_{ij}$ . Також досліджуються різні способи формування таблиць розбіжностей для розрахунку показників каппа (2). Відповідно першого способу,  $C_1$ , враховуються відомі попередні результати класифікації за набором  $X$ , відповідно способу  $C_2$  такі дані не використовуються, контролюються лише отримані результати (шаблони/класи) за МХН. Результати формування таблиць розбіжностей і розрахунків показників каппа (2) за даними табл. 1 для моделі класифікації КМ<sub>1</sub>, коли в якості конкуруючих варіантів розглядалися всі комбінації пар скорочених моделей  $X/\{X_j\}$ , показані табл. 2.

У табл. 2 приведена частина результатів розрахунків оцінок «каппа», яка дозволяє визначити першу змінну для скорочення моделі класифікації –  $X_6$ ,  $K(X_3/X_6) = 0,75$  (значний рівень узгодженості). Для інших не приведених у табл. 2 пар оцінки величин «каппа» були несуттєвими, подібність відсутня, як для  $K(X_1/X_3)=-0,067$ . Можна сказати, що «вплив» фактору  $X_6$  на модель класифікації ураховують та опосередковано представляють фактори  $X_1$ ,  $X_3$  і  $X_4$ . Відзначається, що оцінки подібності за каппа статистикою для таблиць розбіжностей без урахування класифікації за набором  $X$ , (спосіб  $C_2$ ) перевищують чи дорівнюють результати за способом  $C_1$ . У табл. 3 приведено основні результати розрахунків оцінок показників «каппа» для скороченого складу параметрів класифікації,  $(X_1, X_2, \dots, X_5)$ , представлені за схемою табл. 2.

Таблиця 3

$K(X_1/X_2) = -0,600$	<table border="1"> <tr><td></td><td>Так</td><td>ні</td></tr> <tr><td>так</td><td>2</td><td>3</td></tr> </table>		Так	ні	так	2	3
	Так	ні					
так	2	3					

$$K(X_1/X_3) = 0,529$$

	Так	ні
так	3	2
ні	0	3

ні	3	0
----	---	---

$$K(X_1/X_4) = 0,059$$

	Так	ні
так	2	3
ні	1	2

$$K(X_1/X_5) = -0,111$$

	Так	ні
так	1	4
ні	1	2

Найвище значення «каппа» Коена  $K(X_1/X_3) = 0,53$  (помірний рівень узгодженості), тому видаляється  $X_3$  (подібність до  $X_1, X_4, X_5$ ). Відзначається також відмінність результатів стосовно етапів табл. 2 та табл. 3 за способами формування таблиць розбіжностей  $C_1$  і  $C_2$ . А саме – щодо врахування попередньо відомих результати класифікації за набором  $X$ . У табл. 3 більш високі оцінки показників були отримані за способом  $C_1$ , а не у табл. 2. Тобто необхідно розглядати обидва способи  $C_1$  та  $C_2$  при виконанні скорочення числа параметрів моделі, обираючи рішення за більшими значеннями «каппа». Також необхідна перевірка тотожності та можливості корегування набору шаблонів моделі класифікації після видалення параметрів  $\{X_j\}$ . У разі утворення кількох однакових шаблонів скороченої моделі  $X/\{X_j\}$  у одному або у кількох класах  $K_i$ , залишається лише один. Якщо такі шаблони виникли в одному класі, залишають тільки один, а в разі належності зазначених зразків до різних класів  $K_i$  видаляються шаблони із класу з більшим числом екземплярів. Наведемо результати процедури редукції для  $KM_2$ , коли оцінюється класифікація табл. 1 кожним із 26 зразків  $L_{ij}$ .

Таблиця 4

Розрахунки  $m=6$  за зразками ( $KM_2, C_2$ )

$$K(X_1/X_2) = 0,425$$

	Так	ні
так	3	2
ні	3	18

Розрахунки  $m=6$  за зразками ( $KM_2, C_1$ )

$$K(X_1/X_2) = 0,494$$

	Так	ні
так	14	4
ні	2	6

$$K(X_1/X_3) = -0,035$$

	Так	ні
так	1	4
ні	5	16

$$K(X_1/X_3) = 0,299$$

	Так	ні
так	10	5
ні	4	7

$$K(X_1/X_5) = -0,169$$

	Так	ні
так	0	5
ні	3	18

$$K(X_1/X_5) = 0,325$$

	Так	ні
так	13	5
ні	3	5

$$K(X_3/X_5) = -0,182$$

	Так	ні
так	0	6

$$K(X_3/X_5) = 0,308$$

	Так	ні
так	11	2

	ні	3	17		ні	7	6
$K(X_5/X_6) = -0,169$		Так	ні	$K(X_5/X_6) = 0,278$		Так	ні
	так	0	3		так	14	4
	ні	5	18		ні	4	4

Параметр  $X_1$  табл. 4 має найвищий показник «каппа»  $K(X_1/X_2) = 0,494$  (помірний рівень узгодженості) і разом з тим високі оцінки подібності для  $K(X_1/X_3) = 0,299$  та  $K(X_1/X_5) = 0,325$ , які перевищують значення інших.  $X_1$  представлений через зазначені змінні, тому  $X_1$  видаляється з моделі. У табл. 4 також більш високі оцінки показників «каппа» отримані за способом  $S_1$ .

В якості моделі класифікації за нечіткими даними розглядається завдання із визначення авторства творів україномовних авторів (ЗАТ) [1, 7]. У ЗАТ на підставі сукупності ознак, які характеризують набори текстів і утворюють класи або зразки моделі, необхідно визначити клас, до якого належить новий текст (закодований вектор ознак). Система ознак у авторів однакова, а число наборів (творів) може бути різним. Також певні дані кодів ознак профілю можуть бути «збуреними» або навіть відсутніми. Встановлено, що формування одного окремого шаблону для кожного автору представляє певну проблему. Тому кожному автору відповідає кілька зразків/шаблонів у формі векторів нечітких величин. Відповідно [7] натеper існують системи «вимірювання» властивостей текстів із понад 60 параметрів/ознак (розмірність класифікації  $m > 60$ ). Вимоги до моделей класифікації ЗАТ виконанні при представленні даних у формі таблиці табл. 1.

У [1, 7] кожному автору україномовного тексту відповідав один узагальнений шаблон, вибір творів до формування проблематичний. Вхідний вектор (твор невідомого автора) кодується встановленим набором нечітких величин, всі зразки творів моделі мають єдину форму відображення властивостей текстів. Для прикладу дослідження авторства україномовних текстів використано дані [1]. Ознаками текстів україномовних авторів ( $K_1$  – І. Багрянний,  $K_2$  – О. Довженко,  $K_3$  – М. Жадан,  $K_4$  – М. Коцюбинський,  $K_5$  – Л. Українка,  $K_6$  – П. Мирний,  $K_7$  – І. Франко. ( $K_8$  - інші)) були такі:  $X_1$  – математичне очікування;  $X_2$  – середнє квадратичне відхилення;  $X_3$  – рекурентність;  $X_4$  – детермінізм;  $X_5$  – середня довжина діагональних ліній;  $X_6$  – дивергенція;  $X_7$  – ентропія;  $X_8$  – завмирання;  $X_9$  – затримки;  $X_{10}$  – середня кількість слів,  $X_{11}$  – середня кількість складів,  $X_{12}$  – середня кількість літер у реченні;  $X_{13}$  – середня кількість складів та  $X_{14}$  – літер у словах. Для авторів ( $K_1, \dots, K_7$ ) на основі десяти творів формувалися шаблони на основі розрахунку областей можливих діапазонів та середніх значень. За творів інших авторів (Т. Шевченко, М. Стельмах

ін.) формувався шаблон  $K_8$  - інші. Величини  $X_1 - X_{14}$  нормувалися до інтервалу  $[0; 1]$ . Для «визначення авторства» на мережу МХН подавалися нормовані значення характеристик твору невідомого автора; Мережа МХН, як правило, вірно встановлювала клас автора твору.

Разом з тим було встановлено, що для певних вхідних творів були визначені спрощені шаблони із 4-х параметрів, які за результатами класифікації відповідали сукупності параметрів-ознак  $X_1 - X_{14}$ . Тож показана необхідність формування шаблонів моделей ЗАТ з урахуванням вимог методу спрощень [4].

Модель класифікації при даних у форматі CF(A) змістовно відповідає завданню менеджменту – відбору кандидата із зазначеної множини (ЗК). Завдання призначення ЗК [8] відоме, має багато моделей і процедур реалізації, в залежності від типу та ознак параметрів, способів їх отримання і оцінювання ін. Особливість ЗК цієї роботи наступна. Вважається, що реалізується процедура відбору «кандидата» за даних його портфолію (переліку робіт у певних проєктах) або резюме. Кожний з кандидатів  $K_i$  має набір завдань  $Z_{ik}$  портфолію  $P_i$ , які описані без попередніх вимог до структури опису  $Z_{ik}$ . Параметри та оцінки за  $Z_{ik}$  можуть бути представлені і текстом/мовна-форма, відрізняються для різних кандидатів  $P_i$ . У тому числі мають оцінки у формі неточних показників ін. Остаточну структуру і значення оцінок щодо представлення  $Z_{ik}$  визначає менеджмент, який встановлює оцінки всіх ознак у формі коефіцієнтів упевненості CF(A) У підсумку шаблони різних кандидатів  $K_i$  у моделі ЗК суттєво неоднорідні, можуть мати «пропуски» в ознаках, число шаблонів кандидатів  $K_i$  різне. Вхідний «еталон» містить всі характеристики, ураховані та означені «менеджментом» при формуванні моделі ЗК за портфолію  $P_i$ . Метою являється визначення шаблону та  $K_i$ , який в найбільше відповідає «еталону», ким виконувалося подібне завдання.

Постановка ЗК структури табл. 1 (для  $K_1, \dots, K_4, \dots, K_{ij}$ ) з даними, закодованими коефіцієнтами CF(A), відзначається таким. Кандидати мають різну кількість зразків, певні зразки не мають деяких ознак (наприклад,  $K_{11}$  і  $K_{12}$  не мають даних для  $X_5$  і  $X_{11}$ , а  $K_{41}$  не містить  $X_5, X_6, X_9$  та  $X_{11}$ ). У «еталон» включені наступні ознаки: 1) Пріоритет задачі що виконувалась. 2) Складність задачі. 3) Оцінки навичок категорії  $H_1$ . 4) Оцінки навичок категорії  $H_2$ . 5) Оцінка рівню певного фаху 6) Оцінка досвіду виконавця таких завдань. 7) Оцінка рівня визначених знань  $Z_1$ . 8) Завантаженість у проєкті. 9) Навичка  $H_9, \dots$  11) Оцінка навичок контролю. Засобами МХН [1, 8], а також процедурами редукції та каппа Коена, необхідно встановити шаблон індивідуальних ознак  $Z_{ik}$  «виконавця, клас  $K_i$ » у форматі CF(A) для  $(X_1, \dots, X_{11})$ , що найкраще відповідає «еталону», наприклад,  $(X_1=0.9, X_2=0.75, X_3= - 0.5, \dots, X_6= 1, \dots, X_{11}= - 0.3)$ . Структура простору моделі класифікації формується окремо для кожного «еталону вимог», щоб забезпечити встановлену достовірність результату класифікації.

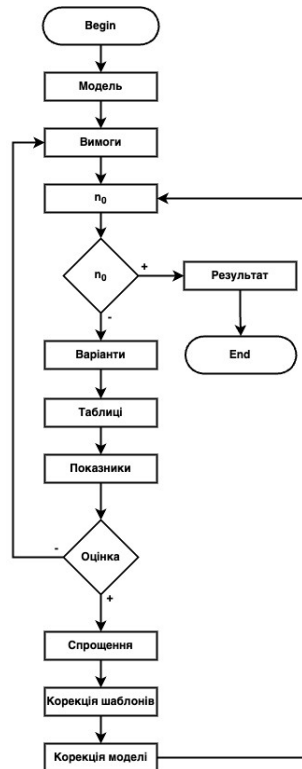


Рисунок 1. Блок-схема алгоритму процедури редукції розмірності моделі

За схемою рис. 1 узагальнено алгоритм редукції на основі каппа статистики виконується таким чином: отримується Модель завдання класифікації за умов невизначеності даних; визначаються Вимоги щодо забезпечення точності та достовірності очікуваних результатів класифікації; за отриманими даними розраховується гранична розмірність моделі  $n_0$ ; перевіряється відповідність поточної моделі класифікації вимогам до моделі за значенням  $n_0$ ; в разі виконання вимог – маємо достовірний Результат; у разі невиконання вимог починається аналіз простору і формування конкуруючих Варіантів спрощених моделей класифікації; для всіх сформованих варіантів реалізації конкуруючих моделей формуються таблиці розбіжностей; за таблицями розраховуються показники статистики каппа Коена та визначається загальна оцінка всіх варіантів; якщо така оцінка схожості конкуруючих моделей не відповідає вимогам достовірності, необхідно змінити вимоги; інакше виконується процедура спрощення (видалення певної структури параметрів моделі класифікації); при тому перевіряється необхідність і виконується корегування систем шаблонів і моделі класифікації.

Розроблено програмний комплекс, призначений для побудови математичної моделі процесів нечіткої класифікації – шаблонів баз знань для реалізації класифікації, розрахунків процесів класифікації на основі мережі МХН. Комплекс забезпечує автоматизацію обробки завдань нечіткої та CF(A) класифікації, щодо

визначення класів вхідних елементів на основі їх неточно визначених ознак, прогнозування категорії «виконавця» та автору україномовного твору.

**Висновки.** У статті отримані удосконалені математичні моделі, алгоритми та програмні засоби, призначені для підвищення достовірності результатів класифікації при невизначених даних, представлених нечіткими величинами та коефіцієнтами упевненості  $CF(A)$ . Удосконалення моделей класифікації з наведеними властивостями даних забезпечується шляхом модифікації нейронних мереж Хеммінга, а також застосування процедур редукції і статистики каппа Коена, які дозволяють сформувати математичні моделі зі встановленими ймовірнісними вимогами до результатів. В роботі запропоновані та реалізовані нові постановки, математичні моделі, а також виконано реалізацію моделей класифікації за нечіткими даними, що вирішують завдання із встановлення авторів україномовних текстів, а також завдання вибору кандидата за моделлю даних у форматі коефіцієнтів  $CF(A)$ . Розроблено програмні засоби для формування моделей класифікації при невизначених даних процедурами редукції і каппа статистики.

Числові експерименти підтвердили достовірність результатів класифікації, а також ефективність запропонованої процедури редукції розмірності моделей на основі статистики каппа Коена.

## ЛІТЕРАТУРА

1. Скалозуб В. В., Горячкін В. М., Клименко І. В., Терлецький І. А., Терленко А. П. Дослідження процедур мережі Хеммінга для управління сервісними системами при неточно визначених і природомовних даних. Наука та прогрес транспорту. 2022. № 3-4 (99-100). С. 33–47. DOI: <https://doi.org/10.15802/stp2022/276411> (in Ukrainian)
2. Великоіваненко Г. І. Оцінювання рівня економічної безпеки на підґрунті відстані Хеммінга. 2018. URL: <https://core.ac.uk/download/pdf/197269753.pdf>
3. Васильев В. И. Индукция и редукция в проблемах экстраполяции. Кибернетика и вычислительная техника. 1998. Вып. 116. С. 65–81.
4. Колесник А. С., Хайрова Н. Ф. Обґрунтування використання статистики каппа Коена в експериментальних дослідженнях NLP Text Mining. Кибернетика та системний аналіз. Т. 58, № 2. 2022. С. 143–153.

5. Li Min Fu, Shortliffe E. H. The application of certainty factors to neural computing for rule discovery. *IEEE Transactions on Neural Networks*. 2000. Vol. 11. Iss. 3. P. 647–657. DOI: <https://doi.org/10.1109/72.846736>
6. Скалозуб В. В., Горячкін В. М., Терлецький І. А. Багатопараметричні інтелектуальні процедури діагностування за неповними і збуреними даними // *Логістика і транспортна безпека: Проблеми та перспективи розвитку в контексті аналізу сучасних викликів і загроз: матеріали доповідей II Міжнародної науково-практичної конференції*. — Дніпро: Середняк Т.К., 2023. С. 42 – 47.
7. Шинкаренко В. І., Демидович І. М. Визначення ознак авторства природномовних текстів. *Штучний інтелект*. 2018. № 3. С. 27–35.
8. Richard A. Brualdi. *Combinatorial matrix classes*. — Cambridge: Cambridge University Press, 2006. — (Encyclopedia of Mathematics and Its Applications). — ISBN 0-521-86565-4
9. Leszek Rutkowski *Metody i techniki sztucznej inteligencji*. Naukowe PWN, Warszawa, 2005. – 520 p.
10. Haykin S. *Neural networks: A Comprehensive Foundation*. Prentice hall: New Jersey, 1999. 1103 p.

#### REFERENCES

1. Skalozub V.V., Goryachkin V.M., Klymenko I.V., Terletsky I.A., Terlenko A.P. Doslidzhennia protsedur merezhi khemminha dlia upravlinnia servisnymy systema-my pry netochno vyznachenikh i pryrodomovnykh danykh. 2022. No. 3-4 (99-100). P. 33–47. DOI: <https://doi.org/10.15802/stp2022/276411>
2. Velykoivanenko, H. I. (2018). Otsiniuvannia rivnia ekonomichnoi bezpeky na pidgrunti vidstani Khemminha. Retrieved from <https://core.ac.uk/download/pdf/197269753.pdf> (in Ukrainian)
3. 1. Vasilev, V. I. (1998). Induktsiya i reduktsiya v problemakh ekstrapolyatsii. *Cybernetics and Computer Engineering*, 116, 65-81. (in Russian).
4. Kolesnyk A. S., Khairova N. F. . Obgruntuvannia vykorystannia statystyky kappa Koena v eksperymentalnykh doslidzhenniakh NLP Text Mining *Cybernetics and system analysis*. Vol. 58, No. 2. 2022. P. 143–153.
5. Li Min Fu, Shortliffe E. H. The application of certainty factors to neural computing for rule discovery. *IEEE Transactions on Neural Networks*. 2000. Vol. 11. Iss. 3. P. 647–657. DOI: <https://doi.org/10.1109/72.846736>
6. Skalozub V.V., Goryachkin V.M., Terletskiy I.A. Bahatoparmetrychni intelektualni protsedury diahnostuvannia za nepovnymy i zburenyemy danymy// *Logistics and transport safety: Problems and prospects of development in the context of analysis of modern challenges and threats: materials of reports II International scientific and practical conference*. — Dnipro: Serednyak T.K., 2023. P. 42-47.

7. Shinkarenko V. I., Demidovych I. M. Determination of signs of authorship of natural language texts. *Artificial Intelligence*. 2018. No. 3. P. 27–35.
8. Richard A. Brualdi. *Combinatorial matrix classes*. — Cambridge: Cambridge University Press, 2006. — (Encyclopedia of Mathematics and Its Applications). — ISBN 0-521-86565-4
9. Leszek Rutkowski *Metody i techniki sztucznej inteligencji*. Naukowe PWN, Warsaw, 2005. – 520 p.
10. Haykin S. *Neural networks: A Comprehensive Foundation*. Prentice hall: New Jersey, 1999. 1103 p.

Скалозуб Владислав Васильович - професор, каф. «Комп'ютерні інформаційні технології», Український державний університет науки і технологій, УДУНТ.

Горячкін Вадим Миколайович – зав. каф. «Комп'ютерні інформаційні технології», Український державний університет науки і технологій, УДУНТ.

Терлецький Ігор Андрійович – аспірант каф. «Комп'ютерні інформаційні технології», Український державний університет науки і технологій, УДУНТ.

Дудник Ілля Петрович – магістрант каф. «Комп'ютерні інформаційні технології», Український державний університет науки і технологій, УДУНТ.

V. V. Skalozub – professor, dep. “Computer and Information Technology”, Ukrainian State University of Science and Technology, USUST.

V. M. Horiachkin – Head of dep. “Computer and Information Technology”, Ukrainian State University of Science and Technology, USUST.

I. A. Terlitskyi – post-graduate student, Dep. “Computer and Information Technology”, Ukrainian State University of Science and Technology, USUST.

I. P. DUDNYK – graduate student, Dep. “Computer and Information Technology”, Ukrainian State University of Science and Technology, USUST

Problems and procedures of classification and diagnosis under conditions of incomplete certainty of the initial data (disturbance, partial omissions etc.) are quite common in practice. One of the main problems of such classification tasks is establishing the property of reliability and completeness of models (choice of data formats, structure and number of samples etc., which provide reliable results). One of the relevant new aspects of diagnostic models is the consistency of the probabilistic requirements for the reliability of the results, the dimensionality of the template parameters, as well as the number of data (classes) according to which the diagnosis is performed. To expand the scope of application

of classification models based on associative memory, as well as to increase the reliability of classification results, we proposed a modified Hamming network.

The article is devoted to the development of mathematical models for the classification of uncertain data represented by fuzzy values and certainty factors  $CF(A)$ . Diagnostic pattern formation procedures use modified Hamming networks, as well as reduction methods and Cohen's kappa statistics. At the same time, the limiting dimensions and composition of the parameters of the classification model are determined, which ensure the established probabilistic requirements for the reliability of the calculation results. The procedure for reducing the model space for diagnosing uncertain data is presented. The article presents formulations, mathematical models, and implementations of classification tasks based on non-deterministic data. An example of a classification model based on fuzzy data is the task of identifying the authors of Ukrainian-language texts. The classification task for data in  $CF(A)$  format corresponds to candidate selection. The results of the numerical modeling made it possible to establish the effectiveness, reliability and efficiency of the proposed procedures for the formation of reliable classification models with uncertain data.

**Keywords:** classification, reliable models, dimensionality of space, fuzzy values,  $CF(A)$  certainty factors, modified Hamming network, reduction procedure, Cohen's kappa statistic, Ukrainian-language texts, author's definition, computer simulation.