

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук  
(повна назва)

Кафедра Штучного інтелекту  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Дослідження особливостей експертних систем для оптимізації  
електропостачання під час війни  
(тема)

Виконав:  
студент 2 курсу, групи СШМ-22-3  
Кучер А.О.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

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

Освітня програма Системи штучного інтелекту  
(повна назва спеціалізації)

Керівник проф. Філатов В.О.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

В.О. Філатов  
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_  
(повна назва)  
Кафедра \_\_\_\_\_ Штучного інтелекту \_\_\_\_\_  
(повна назва)  
Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_  
Спеціальність \_\_\_\_\_ 122 Комп'ютерні науки \_\_\_\_\_  
(код і повна назва)  
Тип програми \_\_\_\_\_ освітньо-наукова \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)  
Освітня програма \_\_\_\_\_ Системи штучного інтелекту \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:  
Зав. кафедри \_\_\_\_\_  
(підпис)  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Кучер Анастасії Олександрівні \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Дослідження особливостей експертних систем для оптимізації електропостачання під час війни \_\_\_\_\_

затверджена наказом університету від 1 квітня 2024 р. № 260Ст

2. Термін подання студентом роботи до екзаменаційної комісії 11 червня 2024 р.

3. Вихідні дані до роботи Розробка програмного додатку з використанням мови програмування php, методи створення експертних систем, методи побудови та обходу графа.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1) Аналіз предметної галузі та постановка задачі

2) Експертні системи в задач електроенергетики

3) Розробка моделі експертної системи

4) Розробка та реалізація алгоритму

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.04.2024	виконано
2	Інтелектуальні інформаційні системи	03.04.2024	
3	Види інтелектуальних систем	03.04.2024	
4	Експертні системи	05.04.2024	
5	Переваги експертних систем	15.04.2024	
6	Використання експертних систем	15.04.2024	
7	Властивості експертних систем	15.04.2024	
8	Етапи створення експертних систем	17.04.2024	
9	Компоненти експертної системи	26.04.2024	
10	Представлення знань в експертній системі	26.04.2024	
11	Етап прийняття рішення	30.04.2024	
12	Створення формального визначення логіки	03.05.2024	
13	База знань, фрейми	06.05.2024	
14	Алгоритм обходу графа розподіленої електричної мережі	06.05.2024	
15	Розробка та реалізація алгоритму	20.05.2024	
16	Розробка вдосконаленого алгоритму Дейкстри	22.05.2024	
17	Обрані технології для реалізації	22.05.2024	
18	Експертна система, тестування	23.05.2024	
19	Захист перед ЕК	11.06.2024	

Дата видачі завдання 1 квітня 2024 р.

Студент Кучер А.О.  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 89 с., 22 рис., 1 дод., 28 джерел.

АЛГОРИТМ ДЕЙКСТРИ, БАЗА ЗНАНЬ, ЕКСПЕРТ, ЕКСПЕРТНІ ЗНАННЯ, ЕКСПЕРТНІ СИСТЕМИ, ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА, ПРЕДСТАВЛЕННЯ ЗНАНЬ, ОБЛАСТЬ ЕКСПЕРТНОСТІ, PHP, SQL.

Об'єкт дослідження – експертна система для оптимізації електропостачання.

Предмет дослідження – аналіз та обробка даних щодо роботи електричних мереж з використанням експертної системи.

Мета дослідження – розробка експертної системи для виконання завдань з обробки та аналізу даних стосовно змін станів роботи станцій електричних мереж.

Методи розробки – розробка програмного додатку з використанням мови програмування php, методи створення експертних систем, методи побудови та обходу графа.

Область застосування – обробка та аналіз даних роботи станцій електричних мереж з метою автоматизації та прискорення роботи диспетчера для забезпечення електропостачання користувачів та з метою запобігання надзвичайних ситуацій.

## **ABSTRACT**

Master's thesis contains: 89 pp., 22 fig., 1 ann., 28 references.

DIJKSTRA'S ALGORITHM, DOMAIN OF EXPERTISE, EXPERT, EXPERT KNOWLEDGE, EXPERT SYSTEMS, INTELLIGENT INFORMATION SYSTEM, KNOWLEDGE BASE, KNOWLEDGE REPRESENTATION, PHP, SQL.

The object of the research is an expert system for optimizing electricity supply.

The subject of the research is the analysis and processing of data on the operation of electrical networks using an expert system.

The purpose of the research is to develop an expert system for performing data processing and analysis tasks related to changes in the operating conditions of power grid stations.

Development methods – development of a software application using the php programming language, methods of creating expert systems, methods of graph construction and traversal.

The field of application is the processing and analysis of data on the operation of power grid stations in order to automate and speed up the work of the dispatcher to ensure power supply to users and to prevent emergency situations.

## ЗМІСТ

Вступ .....	8
1 Аналіз предметної галузі та постановка задачі .....	9
1.1 Аналіз предметної галузі .....	9
1.2 Постановка задачі .....	10
1.3 Інтелектуальні інформаційні системи .....	11
1.4 Види інтелектуальних систем .....	13
1.5 Переваги експертних систем .....	23
1.6 Використання експертних систем .....	25
1.7 Властивості експертних систем .....	26
1.8 Етапи створення експертних систем .....	28
1.9 Види задач .....	29
1.10 Компоненти експертної системи .....	32
1.11 Представлення знань в експертній системі .....	33
1.12 Обмеження у використанні ЕС .....	34
2 Експертні системи в задач електроенергетики .....	36
2.1 Етап прийняття рішення .....	37
2.2 Створення формального визначення логіки оперативних перемикань у системах електропостачання .....	38
3 Розробка моделі експертної системи .....	41
3.1 База знань .....	41
3.2 Модель представлення знань .....	45
3.3 Продукційна модель .....	53
3.4 Вибір СУБД для експертної системи .....	56
3.5 Вибір програмного середовища розробки .....	62
3.6 Алгоритм обходу графа розподіленої електричної мережі .....	68
4 Розробка та реалізація алгоритму .....	74
4.1 Розробка вдосконаленого алгоритму Дейкстри .....	74
4.2. Обрані технології для реалізації .....	80

4.3. Експерта система .....	80
4.4 Тестування.....	82
Висновки.....	85
Перелік джерел посилання.....	86
Додаток А Відомість кваліфікаційної роботи.....	89

## ВСТУП

В сучасному світі людство вже настільки звикло до присутності електроенергії у своєму житті, що при її відсутності зупиняється майже все. Нажаль, в реаліях війни, це не щось неймовірне, а частина нашого життя з відсутністю електропостачання. Звичайні природні катаклізми також можуть стати причиною негаразд. Така проблема може завдати великої шкоди, в першу чергу підприємствам.

За управління розподільними електричними мережами зазвичай відповідає персонал диспетчерів, першочерговою задачею яких є швидке виявлення несправностей та вирішення проблеми з відсутністю електроенергії. До них надходить інформація щодо несправностей, яка повинна бути максимально точною та місткою, адже ці фактори максимально впливають на прийняті рішення щодо вирішення проблем. Проаналізувавши потреби диспетчерів та враховуючи можливості сучасних технологій можна впевнено заявити, що виконання диспетчерами поставлених задач можна успішно оптимізувати та діджиталізувати, що значно скоротить час прийняття рішень. В цьому допоможуть експертні системи, які також зменшать вірогідність помилок.

Першочерговою задачею диспетчера є побудова альтернативного способу постачання електроенергії до місця аварії. На вибір такого маршруту впливають такі фактори як розташування мереж; пріоритетність місць, що потребують постачання альтернативної енергії; навантаження; відсутність повторної подачі електроенергії та інше. Проаналізувавши всі фактори можна впевнено сказати, що помилка диспетчера може призвести до страшних наслідків.

Цінність такого підходу полягає в інваріантності стосовно мереж та здатності навчатися.

Новизна полягає в моделюванні інтелектуальної діяльності диспетчера щодо прийняття рішень.



## 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

Керування розподільчими електричними мережами зазвичай покладається на диспетчерів. Їхня основна відповідальність – швидке виявлення несправностей та вирішення проблеми з відсутністю електроенергії. Точність та чіткість інформації про несправності є важливими, оскільки це суттєво впливає на процеси прийняття рішень. Аналізуючи потреби диспетчерів та використовуючи сучасні технології, можна стверджувати, що їх роботу можна успішно оптимізувати та цифровізувати, скорочуючи час на прийняття рішень. Експертні системи можуть в цьому допомогти, що додатково зменшить ймовірність помилок.

Однією з ключових задач диспетчера є встановлення альтернативних маршрутів постачання електроенергії до місця аварії. Фактори вибору таких маршрутів включають розташування мереж, пріоритетність місць, що потребують енергії, навантаження, відсутність можливості повторної подачі електроенергії та інші. Аналіз усіх цих факторів є важливим, оскільки помилка в прийнятті рішення може призвести до серйозних наслідків.

### 1.1 Аналіз предметної галузі

Диспетчери повинні обробляти достатньо велику кількість вхідної інформації, адже вони повинні враховувати вже існуючі шляхи подачі енергії і оновлену інформацію щодо вже підключених місць після аварії, щоб уникнути повторного підключення, що могло б призвести до нової аварії. Тому є необхідність їм допомогти обробляти вхідну інформацію та обрати правильне рішення.

Для вирішення існуючої проблеми спочатку необхідно проаналізувати та описати інтелектуальну діяльність диспетчера щодо аналізу проблеми та її шляху вирішення, а саме пошук альтернативного

шляху підключення місця аварії до електромережі. Це необхідно для подальшого моделювання.

Далі експертну систему необхідно наповнити даними для подальшого моделювання аварійних ситуацій. Це необхідно для побудови правильної логіки послідовності необхідних дій під час перепідключення електроенергії та для контролю і виключення повторного підключення, яке може призвести до повторної аварійної ситуації на місці вже існуючої аварії.

Щоб експертною системою було зручно користуватися, а головне ефективно та швидко, має значний вплив інтерфейс системи. Всі функції повинні бути розташовані та позначені так, щоб кожен диспетчер інтуїтивно міг знайти її в найкоротший час, адже від цього залежить швидкість реагування на проблему та час її вирішення.

Мережу електропостачання можна представити у вигляді графів та використати математичну логіку.

Наукова новизна полягає у формалізації вибору процесів, які надалі будуть моделювати поведінку диспетчера, якому необхідно проаналізувати ситуацію та прийняти швидке і ефективно рішення щодо перепідключення місць аварії до альтернативних шляхів електропостачання. Також новизна полягає у реалізації інтелектуальних процесів експертної системи з розробленими методами обчислення та аналізу станів електричних мереж, а саме: місця з відсутнім підключенням електроенергії, обробка досить великих обсягів даних, автоматизація процесу аналізу та прийняття рішення з можливістю втручання та управління процесами безпосередньо самими диспетчерами. Практична цінність полягає у можливості подальшого навчання такої системи.

## 1.2 Постановка задачі

Необхідно розробити інтелектуальну експертну систему з можливістю аналізу аварійної ситуації, яка виникла та пропозиції її

вирішення шляхом обрання альтернативного шляху постачання електроенергії враховуючи вже існуючі підключені лінії та виключаючи можливість повторної подачі електроенергії.

### 1.3 Інтелектуальні інформаційні системи

Інтелектуальні інформаційні системи (ІС) – це складні комп'ютерні системи, які об'єднують в собі обробку інформації та штучний інтелект з метою вирішення складних завдань. Ці системи використовують розумові та самонавчальні алгоритми, щоб аналізувати, розуміти та приймати рішення на основі великої кількості даних.

Основні характеристики інтелектуальних інформаційних систем включають:

- аналітичні можливості: ІС можуть використовувати аналітичні інструменти для обробки та аналізу великих обсягів даних, виявлення залежностей та виведення корисної інформації;
- інтелектуальна обробка даних: ці системи володіють здатністю розуміти, класифікувати та інтерпретувати інформацію, подібно до способу, як це робить людський розум;
- самонавчання: деякі ІС мають можливість самонавчання, що дозволяє їм адаптуватися до нових умов та вдосконалювати свої алгоритми на основі накопичених даних;
- підтримка прийняття рішень: ІС можуть допомагати в прийнятті рішень, аналізуючи інформацію та рекомендуючи оптимальні варіанти дій;
- взаємодія з користувачами: деякі ІС володіють інтерфейсами, що дозволяють взаємодіяти з користувачами та враховувати їхні потреби та вхідну інформацію.

Інтелектуальні інформаційні системи використовуються в різноманітних галузях, таких як медицина, фінанси, логістика, наука,

технології та інші, для вирішення завдань, що вимагають складного аналізу та прийняття рішень.

Звичайні ж інформаційні системи мають ряд недоліків, а саме: вони погано адаптуються та підлаштовуються до персоналізованих потреб користувача, а також такі системи досить погано справляються із задачею вирішення задач, які було погано формалізовано. Саме тому перевага надається саме ІС.

Інтелектуальні інформаційні системи (ІС) можна класифікувати за різними критеріями, такими як функціональність, область використання та характеристики. Нижче наведено деякі основні аспекти класифікації ІС:

1) за функціональністю:

- системи обробки природної мови (Natural Language Processing, NLP): використовуються для розуміння та генерації людської мови;
- системи експертних знань: моделюють та використовують експертні знання для прийняття рішень у конкретній області;
- системи машинного навчання: засновані на алгоритмах машинного навчання для аналізу даних та вивчення закономірностей;
- системи розпізнавання образів: використовують алгоритми для розпізнавання та класифікації образів або об'єктів;
- системи планування та прийняття рішень: допомагають в процесі планування та оптимізації рішень в різних ситуаціях;

2) за областю використання:

- медичні ІС: для діагностики, обробки медичних зображень та виконання різних завдань в сфері охорони здоров'я;
- фінансові ІС: для аналізу ринків, управління ризиками та прийняття рішень в фінансовій сфері;
- логістичні ІС: для оптимізації логістичних процесів та управління ланцюгом постачання;

- системи управління знаннями: для зберігання, обробки та використання знань у певній галузі;

3) за характеристиками:

- системи з підтримкою прийняття рішень: допомагають у прийнятті оптимальних рішень на основі аналізу даних;

- системи автоматизованого планування: використовуються для автоматизації процесів планування та розподілу ресурсів;

- системи моніторингу та прогнозування: для відстеження подій, моніторингу змін та прогнозування майбутніх станів.

Ці класифікації можуть перетинатися, і багато ІС можуть використовувати комбінації різних функціональних підходів в залежності від конкретних завдань та областей використання.

#### 1.4 Види інтелектуальних систем

Інтелектуальні системи охоплюють широкий спектр технологій та підходів, включаючи різні види систем для різних завдань. Ось деякі основні види інтелектуальних систем:

- експертні системи (ES): використовуються для моделювання та використання експертних знань у конкретній області, допомагають у прийнятті рішень та вирішенні проблем;

- системи машинного навчання (ML): засновані на алгоритмах, які дозволяють системам вивчати та покращувати свою продуктивність на основі даних без явного програмування;

- системи обробки природної мови (NLP): дозволяють комп'ютерам розуміти, інтерпретувати та генерувати людську мову;

- системи комп'ютерного зору: використовуються для розпізнавання образів та об'єктів на зображеннях або відео;

- системи автоматизованого планування: використовуються для оптимізації планування та вирішення завдань розподілу ресурсів;
- системи розпізнавання мови: дозволяють комп'ютерам розпізнавати та інтерпретувати голосові команди або текстові дані;
- системи розпізнавання образів та голосу: комбінують функції розпізнавання образів та голосового введення для взаємодії з користувачем;
- системи імітації та моделювання: використовуються для створення віртуальних середовищ та моделювання різних сценаріїв;
- системи діагностики та моніторингу: використовують аналітичні методи для виявлення аномалій або відстеження стану об'єктів або систем;
- системи автоматизованого аналізу даних: використовують алгоритми для аналізу великих обсягів даних та виділення важливої інформації.

Ці види інтелектуальних систем можуть взаємодіяти між собою або використовуватися окремо в залежності від конкретних вимог завдань та областей використання.

Розглянемо більш детально кожен вид інтелектуальної системи.

#### 1.4.1 Експертні системи

Експертні системи (Expert Systems, ES) – це тип інтелектуальних систем, які моделюють та використовують експертні знання у конкретній області для вирішення складних завдань та прийняття рішень. Ці системи намагаються емулювати процес прийняття рішень, який виконується людьми-експертами, застосовуючи формалізовані правила та логічні алгоритми.

Основні компоненти експертних систем включають:

- база знань (Knowledge Base, KB): це основний елемент системи, в якому зберігається експертне знання. Знання представляється у вигляді правил, фактів, процедур, ієрархій або інших структур;
- механізм управління знанням (Inference Engine): відповідає за використання знань для прийняття рішень. Цей механізм використовує логічні алгоритми та правила для виведення нової інформації з наявного знання;
- система введення (User Interface): забезпечує взаємодію користувача з експертною системою. Включає інтерфейс для введення даних, запитань та виведення результатів;
- система виведення (User Interface): відповідає за представлення результатів та пояснення прийнятих рішень користувачеві;
- база фактів (Fact Base): зберігає фактову інформацію, яка може бути використана для виведення нових знань.

Процес роботи експертної системи може бути узагальнений наступним чином:

- запитання: система задає користувачеві ряд запитань для збору необхідної інформації;
- введення даних: користувач вводить або підтверджує інформацію, необхідну для прийняття рішення;
- виведення: використовуючи базу знань та механізм виведення, система приходять до висновків та надає відповіді;
- пояснення: система може пояснити свої висновки, надаючи аргументи та логіку, які призвели до конкретного результату.

Експертні системи застосовуються в різноманітних галузях, таких як медицина, фінанси, інженерія, управління та інші, де важливо використовувати експертне знання для ефективного розв'язання проблем.

Експертні системи використовують різні способи обробки знань для прийняття рішень та вирішення проблем у визначених областях. Основні методи обробки знань в експертних системах включають:

1) символічне представлення знань:

- правила: формалізовані правила, які визначають логіку та взаємодію між фактами. Правила виглядають як «Якщо [умова], то [висновок]»;

- фрейми: структури, що містять інформацію про конкретні об'єкти чи концепції та їх властивості;

2) база знань:

- системи представлення знань: використовуються для організації та зберігання знань у вигляді фактів, правил та ієрархій;

- логічні та математичні моделі: використовуються для вираження відносин та закономірностей між різними елементами знань;

3) механізми виведення (Inference Engines):

- пряма логіка (Forward Chaining): система виводить нові факти на основі вже відомих правил та вхідних даних;

- зворотна логіка (Backward Chaining): система спрямовується на досягнення певної мети та виводить відповідні правила для досягнення цієї мети;

4) керування нечіткою інформацією:

- логіка нечіткої логіки (Fuzzy Logic): дозволяє обробляти та враховувати нечітку, невизначену інформацію, що може відображати реальний світ;

5) системи машинного навчання:

- навчання з учителем (Supervised Learning): система вивчає правила та знання на основі тренувальних даних зі звичайними прикладами;

- навчання без учителя (Unsupervised Learning): система аналізує дані без наперед визначених відповідей, шукаючи приховані структури та взаємозв'язки;

6) імітація роботи експертів:



- методи вивчення експертних рішень: деякі системи намагаються моделювати процеси та рішення, які приймаються людьми-експертами;

7) механізми вибору та виправлення:

- механізми вибору стратегій: визначають, як система обирає конкретний метод розв'язання задачі;

- системи виправлення помилок: включаються для виявлення та виправлення неправильних або невірних рішень.

Ці методи можуть використовуватися окремо чи в поєднанні, залежно від конкретних потреб та характеристик експертної системи.

#### 1.4.2 Системи машинного навчання

Системи машинного навчання (МН) – це галузь штучного інтелекту, що дозволяє комп'ютерам вчитися та покращувати свою продуктивність без явного програмування. Машинне навчання використовує алгоритми та статистичні моделі для навчання систем визначати закономірності в даних, вирішувати завдання та робити прогнози.

Машинне навчання прийнято розділяти на наступні види.

Навчання з учителем (Supervised Learning). Система отримує тренувальний набір даних, що містить вхідні та вихідні пари, та вчиться прогнозувати вихід для нових вхідних даних (рисунок 1.1).

В даній схемі «вчителю» відома інформація про зовнішнє середовище, задана у вигляді послідовності або пакета вхідних векторів, а також «правильна реакція» на ці сигнали, представлена у вигляді навчального сигналу. Природно, що реакція ненавченої мережі відрізняється від «правильної» реакції вчителя, в результаті чого виникає помилка. У процесі навчання необхідно так налаштувати параметри ШНМ, щоб деяка скалярна функція від помилки (критерій якості) досягла свого мінімального значення. Навченою вважається мережа, яка в деякому, як правило, статистичному

сенсі повторює реакцію вчителя. Оскільки інформація про зовнішнє середовище зазвичай має нестационарний характер, процес навчання йде безперервно, для чого використовуються ті чи інші рекурентні процедури.

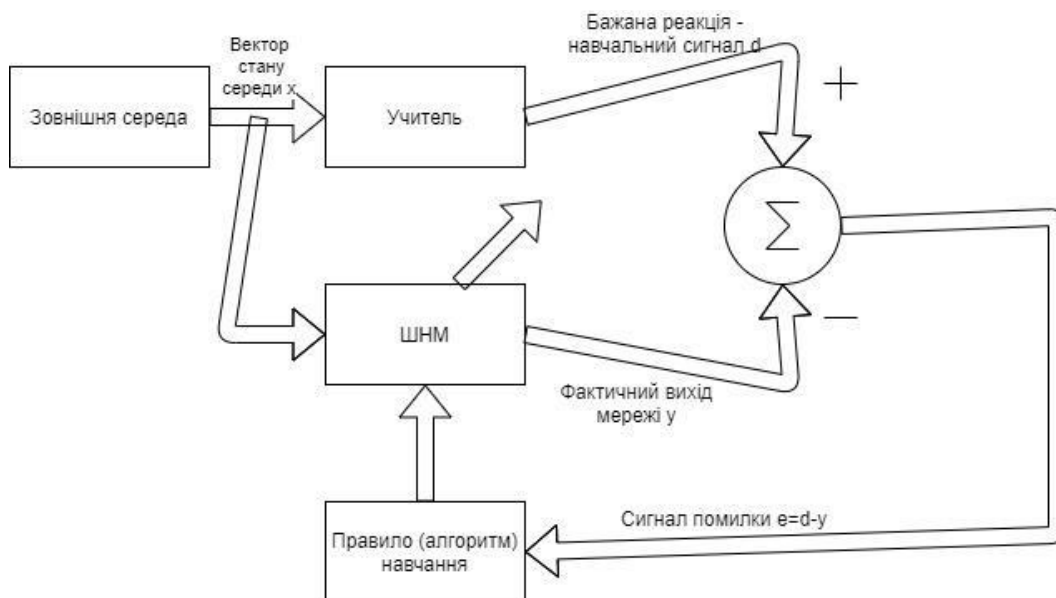


Рисунок 1.1 – Схема навчання з учителем

Даний метод доцільно використовувати при досить великих обсягах даних. Розглянемо приклад з багатьма фотографіями домашніх тварин з маркерами: це кішка, а це собака. Головною задачею є створення алгоритму, за допомогою якого машина могла б за допомогою фотографії, яку «не бачила» раніше, визначити, хто на ній зображений: кішка або собака. У ролі «вчителя» в даному випадку виступає людина, яка заздалегідь проставила маркери. Машина самостійно обирає ознаки, за якими вона відрізняє кішок від собак.

Завдяки такому методу знаходження алгоритму навчання машини, в подальшому знайдений нею алгоритм може бути також застосовано та переналаштовано на рішення іншої задачі, наприклад, на розпізнавання курей і качок. Машина знову сама виконає складну і копітку роботу по виділенню ознак, за якими буде розрізняти цих птахів. А неймережу, яку

навчили розпізнавати кішок, можна швидко навчити обробляти результати комп'ютерної томографії.

Навчання без учителя або самонавчання (Unsupervised Learning). Система вивчає структуру даних без заздалегідь визначених відповідей, спробуючи знайти приховані патерни або групи.

Альтернативою для навчання з учителем є навчання без учителя або самонавчання. При навчанні без учителя, реакція на сигнали оточуючого середовища невідома. Процес самонавчання представлено на рисунку 1.2.

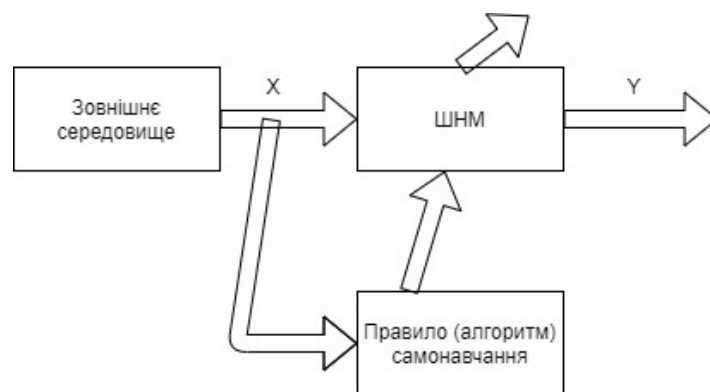


Рисунок 1.2 – Схема навчання без учителя (самонавчання)

Мережі, які реалізують парадигму самонавчання, призначені, як правило, для аналізу внутрішньої латентної структури вхідної інформації і вирішують завдання автоматичної класифікації, кластеризації, факторного аналізу, компресії даних.

Окрім маркованих даних, для яких використовується навчання з учителем, також існує багато, але даних без міток все ж таки більше. Це зображення без підписів, аудіозаписи без коментарів, тексти без анотацій. Завдання машини при неконтрольованому навчанні – знайти зв'язку між окремими даними, виявити закономірності, підібрати шаблони, упорядкувати дані або описати їх структуру, виконати класифікацію даних. Для таких цілей використовується самонавчання. Навчання без учителя використовується, наприклад, в рекомендаційних системах, коли в інтернет-

магазині на основі аналізу попередніх покупок покупцеві пропонуються товари, які можуть зацікавити його з більшою ймовірністю, ніж інші. Або коли після перегляду якогось відеокліпу на порталі YouTube відвідувачеві пропонують десятки посилань на ролики, чимось схожі на переглянутий. Або коли Google у відповідь на один і той же запит ранжує посилання в результатах пошуку для одного користувача інакше, ніж для іншого, оскільки враховує історію пошуків.

Навчання з підкріпленням (Reinforcement Learning). Система навчається, взаємодіючи з навколишнім середовищем і отримуючи відповідь у вигляді винагород або покарання за свої дії.

Навчання з підкріпленням є своєрідною золотою серединою (компромісом) між двома попередніми парадигмами. Частіше за все навчання з підкріпленням плутають з навчанням з заохоченням. Але потрібно ці поняття розрізняти, адже при навчанні з заохоченням доступна лише непряма інформація про правильну реакцію на вхідний сигнал.

На рисунку 1.3 приведена схема процесу навчання з підкріпленням.

Нейронна мережа створює відображення вхідної інформації  $x$  у вихідний вектор  $y$  у вигляді  $y = F(x)$ , проте, оскільки навчальний сигнал  $d$  в явному вигляді не заданий, неможливо отримати помилку  $e = d - y$ , на підставі якої відбувається навчання. Передбачається, що є деякі апріорні знання, що дозволяють зв'язати евристичний сигнал підкріплення  $\tilde{d}$  з неспостережуваних бажаним виходом  $d$  з допомогою деякої функції  $\tilde{F}$ , що відображає  $d$  в  $\tilde{d}$ . Зазвичай ця функція враховує зв'язок вихідних сигналів мережі  $y$  із подіями, які спостерігаються у зовнішньому середовищі, для чого в схему навчання вводиться додатковий блок – «критик» [1], що відображає поведінку мережі в сигнал  $\tilde{y} = \tilde{F}(F(x))$ . Далі обчислюється евристична помилка  $\tilde{e} = \tilde{d} - \tilde{y}$ , на основі якої і реалізується процес навчання.

Процес навчання з підкріпленням розбивається на два відносно незалежних етапи:

- навчання того, як вихідний сигнал мережі впливає на спостережувані змінні середовища, тобто відновлення відображення;
- навчання мережі на основі мінімізації прийнятого критерію.

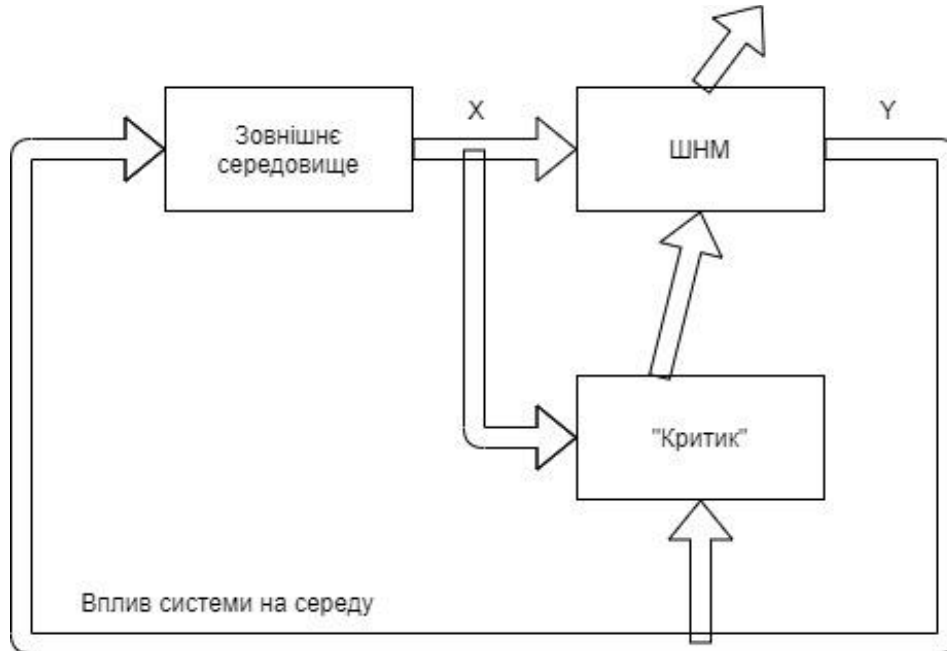


Рисунок 1.3 – Схема навчання з підкріпленням

Ця парадигма тісно пов'язана з ідеями динамічного програмування [2] і в теорії штучних нейронних мереж відома також як нейродинамічне програмування [1].

Досить широке поширення набула також парадигма змішаного навчання, коли частина параметрів мережі налаштовується за допомогою навчання з учителем, а інша частина або архітектура в цілому – за допомогою самонавчання. Цей підхід набув найбільшого поширення при навчанні радіально-базисних ШНМ.

Таке навчання є окремим випадком контрольованого навчання, але вчителем в даному випадку є «середовище». Машина (її в цій ситуації часто називають «агент») не має попередньої інформацією про середовище, але має можливість виконувати в ній будь-які дії. Середина реагує на ці дії і тим

самим надає агенту дані, які дозволяють йому реагувати на них і вчитися. Фактично агент і середовище утворюють систему зі зворотним зв'язком.

Навчання з підкріпленням використовується для вирішення більш складних завдань, ніж навчання з учителем і без вчителя. Воно використовується, наприклад, в системах навігації для роботів, які навчаються уникати зіткнень з перешкодами досвідченим шляхом, отримуючи зворотний зв'язок при кожному зіткненні. Навчання з підкріпленням використовується також в логістиці, при складанні графіків і плануванні завдань, при навчанні машини логічним іграм (покер, нарди, го та інші).

До основних завдань машинного навчання відносять:

- класифікація: призначення вхідних даних до конкретного класу чи категорії;
- регресія: прогнозування числового значення на основі вхідних даних;
- кластеризація: групування схожих елементів у великому наборі даних;
- виявлення аномалій: виявлення незвичайних чи аномальних патернів у даних;
- здатність до генерування: створення нових елементів або даних, що схожі на вхідні.

Основні етапи роботи систем машинного навчання:

- підготовка даних: відбираються, очищаються та обробляються дані для подальшого використання;
- вибір моделі: обирається або створюється відповідна модель машинного навчання;
- тренування моделі: модель «навчається» на тренувальному наборі для визначення оптимальних параметрів;
- оцінка та тестування: модель оцінюється на тестових даних для перевірки її ефективності та генералізації;

– використання та вдосконалення: модель використовується для прогнозування чи вирішення завдань, а також може піддаватися постійному вдосконаленню з новими даними.

Існують такі алгоритми машинного навчання:

– дерева рішень (Decision Trees): використовують деревоподібну структуру для прийняття рішень;

– лінійна регресія (Linear Regression): визначає лінійний зв'язок між вхідними та вихідними даними;

– нейронні мережі (Neural Networks): моделюють взаємозв'язки, навчаючись за аналогією зі структурою людського мозку;

– метод опорних векторів (Support Vector Machines): використовує гіперплощини для класифікації даних.

Зазвичай машинне навчання використовується у медицині (діагностика, прогнозування захворювань та розробка нових ліків), фінанси (прогнозування ринків, управління ризиками та робота фінансових портфелів), технології (розпізнавання образів, мова, автономні автомобілі та інше), електронна комерція (рекомендації для користувачів та персоналізовані пропозиції).

Машинне навчання стає все більш важливою та розповсюдженою галуззю, застосовуючи свої можливості в різних сферах життя та промисловості.

### 1.5 Переваги експертних систем

Експертні системи представляють собою інтелектуальні інформаційні системи, які використовують знання та експертизу для прийняття рішень у конкретній області. В науковому вимірі експертні системи визначаються своєю здатністю до емуляції професійного експерта та забезпечення високої ефективності у вирішенні завдань. Розглядання переваг експертних систем у науковому контексті розкриває їхню значущість та внесок у різні галузі.

Ефективне управління знаннями.

Експертні системи відзначаються здатністю систематизувати та зберігати обширні області експертних знань, що є важливим фактором у наукових дослідженнях. Вони дозволяють ефективно управляти знаннями та роблять їх доступними для подальшого використання.

Автоматизація прийняття рішень.

Експертні системи використовують накопичені експертні знання для автоматизації прийняття рішень у конкретних ситуаціях. Це дозволяє прискорити та удосконалити процеси прийняття рішень в наукових дослідженнях, сприяючи точній та обґрунтованій аналізу.

Експертна ефективність.

Експертні системи можуть емулювати роботу висококваліфікованих експертів у різних галузях. Це забезпечує стабільну та надійну експертну ефективність, що особливо важливо у вимірі наукових досліджень.

Розпізнавання шаблонів та аналогій.

Експертні системи демонструють здатність розпізнавати шаблони та аналогії в інформації, що дозволяє виявляти внутрішні зв'язки та тенденції у складних наукових величинах.

Адаптація до змін.

Здатність експертних систем до самонавчання та адаптації до нових знань інтегрується у наукове середовище, де знання постійно розвивається. Це дозволяє системам підтримувати актуальність та реагувати на зміни у дослідницькій області.

Підтримка прийняття рішень у нечіткому середовищі.

Експертні системи можуть успішно працювати у нечітких умовах та враховувати неоднозначність в наукових дослідженнях, де дані можуть бути неповними чи неоднозначними.

Підвищення продуктивності та ефективності.



Використання експертних систем в наукових дослідженнях призводить до підвищення продуктивності та ефективності в обробці даних, аналізі результатів та прийнятті рішень.

Можливості інтеграції.

Експертні системи можуть легко інтегруватися з іншими технологіями, що відкриває можливості для синергетичного використання з іншими науковими інструментами.

Загалом, враховуючи ці переваги, експертні системи стають значущими інструментами для наукових досліджень, надаючи зручний інструментарій для обробки та використання експертних знань у різних наукових областях.

## 1.6 Використання експертних систем

Експертні системи застосовуються в різних галузях завдяки їхній здатності моделювати та використовувати експертні знання для прийняття рішень. Ось декілька галузей використання експертних систем:

– медицина: експертні системи використовуються для діагностики хвороб, надання рекомендацій по лікуванню, планування операцій та інших медичних завдань. Вони можуть аналізувати медичні дані та рекомендувати оптимальні терапевтичні підходи;

– фінанси: у фінансовому секторі експертні системи використовуються для ризик-аналізу, управління портфелем, прогнозування ринкових тенденцій та надання фінансових консультацій;

– виробництво та інженерія: експертні системи можуть допомагати у розпізнаванні несправностей в обладнанні, плануванні виробництва, оптимізації процесів та вирішенні інженерних завдань;

– системи підтримки прийняття рішень: використовуються для допомоги у прийнятті рішень в різних галузях, включаючи управління бізнесом, освітою та соціальними сферами;

– технічне обслуговування та ремонт: експертні системи можуть допомагати в технічному обслуговуванні, діагностиці несправностей та розробці планів ремонту для різних видів обладнання;

– екологія: в галузі охорони навколишнього середовища експертні системи використовуються для аналізу впливу на довкілля, моніторингу забруднення та вирішення екологічних проблем;

– транспорт та логістика: експертні системи можуть допомагати в розподілі ресурсів, оптимізації маршрутів транспортних засобів та управлінні логістичними процесами;

– освіта: в галузі освіти експертні системи можуть надавати індивідуалізовані навчальні матеріали, визначати слабкі сторони у навчанні та рекомендувати методи вдосконалення;

– автоматизація домашніх систем: в галузі домашнього комфорту експертні системи можуть використовуватися для автоматизації освітлення, опалення, безпеки та інших систем у будинку;

– галузі науки та досліджень: експертні системи використовуються для аналізу даних, прогнозування результатів досліджень, автоматизації аналітичних процесів та інших наукових завдань.

Ці галузі використання демонструють універсальність та розмаїття можливостей експертних систем у вирішенні різноманітних завдань у різних сферах діяльності.

## 1.7 Властивості експертних систем

Експертні системи визначаються рядом характерних властивостей, які роблять їх унікальними та ефективними для вирішення конкретних завдань у різних галузях. Ось деякі характеристики, які є типовими для експертних систем:

– знання експерта: одна з ключових рис експертних систем - це використання знань висококваліфікованих експертів у конкретній галузі. Це знання представлене у вигляді бази правил, фактів та евристичних правил;

– імітація прийняття рішень: експертні системи спроектовані для імітації процесу прийняття рішень, який зазвичай властивий екпертам у певній області. Вони враховують логіку та розсуд експерта при аналізі проблем та прийнятті рішень;

– база знань: експертні системи мають базу знань, яка включає в себе експертні правила, факти та додаткові дані, необхідні для вирішення конкретних завдань. Ця база знань є основою для прийняття рішень;

– інференція: інференція в експертних системах відноситься до процесу виведення нової інформації на основі наявних даних та знань в системі. Вона дозволяє здійснювати логічні висновки та робити припущення для вирішення проблем;

– експліцитне представлення знань: знання в експертних системах представлене явно та структуровано, що дозволяє системі легко інтерпретувати та використовувати його для вирішення певних завдань;

– система обробки знань: експертні системи використовують спеціалізовані алгоритми та методи для обробки знань, такі як логічне виведення, фаззі-логіка, або нейромережеві підходи;

– можливість пояснення рішень: експертні системи можуть надавати пояснення своїх рішень, що робить їх зрозумілими та прийнятними для користувачів. Це важливо, особливо в областях, де важливо розуміти логіку прийняття рішень;

– спрощений доступ до експертної експертизи: експертні системи роблять експертні знання доступними для широкого кола користувачів, що дозволяє ними користуватися без прямого звертання до фахівця;

– можливість навчання: деякі експертні системи мають можливість навчання на основі нових даних чи досвіду, що дозволяє системам адаптуватися до змін в середовищі та оновлювати свою базу знань;

– високий рівень автоматизації: експертні системи призначені для високого рівня автоматизації прийняття рішень, спрощуючи завдання та зменшуючи необхідність в людській інтервенції.

Ці риси роблять експертні системи потужним інструментом для вирішення завдань у різних галузях, де важливо використовувати експертні знання та логіку.

## 1.8 Етапи створення експертних систем

Розробка експертної системи – це комплексний процес, який включає в себе кілька етапів. Нижче подано загальний огляд етапів розробки експертної системи:

– формулювання проблеми: на цьому етапі визначається конкретна проблема чи завдання, яке експертна система буде розв'язувати. Це включає в себе визначення області застосування, визначення мети та об'єкта дослідження;

– збір інформації та знань: на цьому етапі здійснюється збір інформації та експертних знань, які будуть використовуватися в системі. Це може включати в себе інтерв'ю з експертами, аналіз літератури, дослідження існуючих даних та інше;

– формалізація знань: зібрані знання експертів формалізуються у вигляді бази знань, яка може бути легко розуміта та використовувана комп'ютером. Це може включати визначення правил, створення схем, опис об'єктів та їх взаємодії;

– розробка архітектури системи: на цьому етапі визначається архітектура експертної системи. Обираються технології, методи та інструменти для реалізації системи. Також розробляється структура бази знань та логіка інференції;

- розробка імплементації: на цьому етапі програмістами реалізується сама експертна система. Це включає в себе написання програмного коду, розробку інтерфейсу користувача та інші технічні аспекти;

- тестування: після реалізації системи проводяться ретельні тести, щоб переконатися в правильності та ефективності роботи експертної системи. Тести можуть включати в себе введення різних сценаріїв та перевірку правильності виведених результатів;

- впровадження: після успішного тестування система готова до впровадження у реальне виробниче чи бізнесове середовище. Здійснюється її встановлення та навчання користувачів;

- підтримка та апгрейд: етап підтримки передбачає надання технічної підтримки користувачам та вирішення проблем, які можуть виникнути у процесі експлуатації. Також може включати апгрейд системи з метою вдосконалення функціональності чи додавання нових можливостей.

Розробка експертної системи - це ітеративний процес, і під час роботи над проектом може виникнути необхідність повернутися до попередніх етапів для внесення змін чи доповнень. За якістю роботи експертні системи можуть майже повністю прирівнятися до роботи експерта, а в деяких задачах можуть дати навіть кращі результати роботи.

Найголовніше в роботі з експертною системою це якісний та інтуїтивно зрозумілий інтерфейс, адже він значно прискорює та полегшує роботу людини. Завдяки здатності експертної системи нарощувати знання та на відміну від людини, якій важко утримувати велику кількість інформації, вона може ними досить оперативно та якісно оперувати для вирішення задач.

## 1.9 Види задач

Експертні системи можуть вирішувати різноманітні задачі в різних галузях, де потрібна експертна експертиза для прийняття рішень.

Ось деякі типи задач, які експертні системи здатні вирішувати:

– діагностика та вирішення проблем: експертні системи можуть допомагати в ідентифікації причин несправностей, визначенні діагнозу в медицині, техніці чи інших галузях, а також вирішенні проблемних ситуацій;

– прогнозування та аналіз: експертні системи можуть використовуватися для прогнозування різних подій, таких як ринкові тенденції, погодні умови, фінансові результати та інше. Також вони можуть здійснювати аналіз великих обсягів даних;

– управління ресурсами: експертні системи можуть допомагати в оптимізації використання ресурсів, таких як персонал, матеріали, енергія, для забезпечення максимальної ефективності та економії;

– планування та розподіл завдань: в області управління проектами чи виробництва експертні системи можуть визначати оптимальні графіки та розподіл завдань для досягнення найкращих результатів;

– системи підтримки прийняття рішень: експертні системи широко використовуються в системах підтримки прийняття рішень для аналізу ситуацій, визначення альтернатив та рекомендацій для кращих варіантів дій;

– експертні консультанти: експертні системи можуть функціонувати як експертні консультанти, які надають користувачам поради, відповіді на запитання та допомагають у вирішенні конкретних завдань;

– технічне обслуговування та діагностика: в галузі технічного обслуговування експертні системи використовуються для діагностики несправностей, надання рекомендацій щодо ремонту та обслуговування технічного обладнання;

– дизайн та проектування: експертні системи можуть допомагати в дизайні та проектуванні, роблячи рекомендації щодо оптимальних конструкцій, матеріалів чи технологій;

– управління якістю: в галузях виробництва чи послуг експертні системи використовуються для контролю та підтримки високого рівня якості продукції чи послуг;

– оптимізація бізнес-процесів: експертні системи можуть допомагати в аналізі та оптимізації бізнес-процесів, визначаючи оптимальні стратегії та рішення для досягнення бізнес-цілей.

Ці задачі вказують на широкий спектр застосування експертних систем у різних галузях та контекстах.

Здебільшого експертні системи виконують операції, які імітують діяльність експерта, а саме коли необхідно водночас обробити велику кількість вхідної інформації та спираючись на вже набутий попередній досвід прийняти швидко та максимально вигідне рішення ситуації. Тобто експертні системи вирішують саме задачі, які трапляються в реальному житті, а не теоретичні та дослідні ситуації, які необхідні лише для розвитку наукових досліджень. Також програми для вирішення не реальних, а лише дослідницьких ситуацій можуть не бути швидкими та продуктивними, адже вони несуть тільки науково-дослідницький характер.

Експертні системи також ще називають системами, які базуються на знаннях (knowledge-based system). Цей термін вважається більш вузьким на відміну від терміну «Експертні системи». Такі системи базуються на роботі зі знаннями, а не на алгоритмах та статистиках. Кожна експертна система містить людські знання з конкретної області та продовжує їх накопичувати та використовувати у своїй роботі.

Також необхідно враховувати, що не всі створені системи можна назвати експертними, адже експертна система повинна вміти обґрунтувати своє рішення спираючись на знання, які вона містить. Таке вміння також грає важливу роль під час тестування системи, адже таким чином тестувальники зможуть відслідкувати не правильну роботу системи.

## 1.10 Компоненти експертної системи

Експертна система – це інтелектуальна інформаційна система, яка використовує знання та експертні правила для прийняття рішень в певній області. Компоненти експертної системи можна розглядати як будівельні блоки, які взаємодіють для вирішення конкретних завдань.

Ці компоненти є основою для розробки та функціонування експертних систем, які володіють здатністю аналізувати складні ситуації та надавати експертні рекомендації відповідно до вхідних даних та контексту проблеми.

Нижче розглядаються основні компоненти експертної системи:

– база знань (Knowledge Base): база знань є фундаментальним компонентом експертної системи, де зберігається експертна інформація, факти, правила та вивчені шаблони. Ця база знань інтегрує в себе експертний досвід та допомагає системі приймати рішення шляхом використання логічних виводів;

– механізм інференції (Inference Engine): механізм інференції відповідає за обчислення та логічне виведення нової інформації на основі знань, що містяться в базі. Він використовує правила та стратегії для здійснення логічних висновків та визначення оптимальних рішень;

– інтерфейс користувача (User Interface): інтерфейс користувача забезпечує взаємодію між системою та користувачем. Це може бути текстовий інтерфейс, графічний інтерфейс, голосовий інтерфейс або комбінація цих елементів. Інтерфейс надає можливість користувачеві вводити запитання, отримувати рекомендації та взаємодіяти з системою;

– механізм перебору (Search Mechanism): механізм перебору використовується для пошуку та вибору оптимальних шляхів або варіантів рішень в базі знань. Він допомагає експертній системі розглядати різні можливості та знаходити найбільш ефективні рішення;

– механізм навчання (Learning Mechanism): деякі експертні системи мають механізм навчання, який дозволяє системі адаптуватися до нової



інформації, отриманої під час роботи або в результаті користувацького введення. Цей компонент покращує продуктивність системи та робить її більш гнучкою;

– модуль визначення несправностей (Troubleshooting Module): у випадках діагностики експертні системи можуть мати модуль для визначення несправностей, який допомагає в локалізації та ідентифікації причин проблем або несправностей;

– модуль пояснення рішень (Explanation Module): деякі системи обладнані модулем пояснення рішень, який може пояснювати користувачеві логіку прийняття системою рішень. Це допомагає зрозуміти виведені результати та збільшує довіру користувачів.

### 1.11 Представлення знань в експертній системі

Зазвичай знання в експертній системі представляються у формі «If...» «Then...», у вигляді правил (система XCON/R1 – використовується для проектування конфігурації систем комп'ютерних та електричних). Також такий підхід застосовується при проектуванні невеликих систем, які призначені для розв'язання спеціалізованих невеликих завдань застосовуючи невелику кількість правил. Для навчання таких видів систем не обов'язково мати знання експерта, можна використовувати інші джерела знань такі як книги чи статті та інші.

З метою визначення знань, якими система не оперує з певних причин – проводиться діагностика системи за допомогою інтерпретації даних. Використовують підхід найглибшого розкриття області експертності системи. Діагностика несправностей буде визначатися за допомогою сформованих гіпотез, наприклад, «навчена система знає/не знає та вміє/не вміє».

Головним недоліком такого підходу є той факт, що поняття «не вмію» можна по різному трактувати, адже таку відповідь можна отримати або коли

система не знає відповіді, або коли не вистачило часу на пошук відповіді. Щоб уникнути такої ситуації необхідно вдосконалити експертну систему з урахуванням виключення існування такої ситуації як брак часу. Такого результату можна досягти шляхом визначення необхідного часу практичним шляхом, а не навмання.

### 1.12 Обмеження у використанні ЕС

В будь-якому випадку, яка б найкраща експертна система не була, але вона буде мати певні обмеження у використанні на відміну від людини-експерта.

Зазвичай такі системи досить важкі у використанні та без певного досвіду буде досить важко із ними працювати.

Системи, які працюють за принципом питання-відповідь будуть працювати значно повільніше, аніж реальна людина-експерт.

Експертні системи застосовуються в більш вузьких галузях.

Реальна людина-експерт може в певних ситуаціях ще спиратися на свою інтуїцію, на відміну від експертної системи.

Експертні системи ґрунтуються на наявних експертних знаннях, і їх ефективність залежить від якості та обсягу цих знань. Якщо система стикається з новими чи непередбаченими ситуаціями, для яких немає заздалегідь визначених правил, вона може неправильно функціонувати.

Експертні системи можуть мати труднощі у роботі з нечіткою чи неповною інформацією. Якщо правила не визначені чітко чи якщо інформація має непевність, це може призводити до неточностей у виведених рішеннях.

Експертні системи можуть виявити велику стійкість до змін у знаннях або правилах. Вони можуть вимагати періодичного оновлення та налаштування для врахування нової інформації чи змін у домені.

Спроектовані для конкретних завдань, експертні системи можуть мати обмежену адаптивність до нових завдань чи доменів. Вони можуть потребувати значних зусиль для переорієнтації на нові області.

Деякі експертні системи можуть виявити складнощі у поясненні своїх виведених результатів. Це може бути проблемою, особливо у випадках, коли важливо зрозуміти логіку прийняття рішень.

Створення та підтримка експертних систем може бути витратною задачею. Вони вимагають великих зусиль для збору, формалізації та введення експертних знань, а також для забезпечення їхньої актуальності та ефективності в змінюючихся умовах.

У випадках, коли експертні системи використовуються в критичних сферах, таких як медицина чи право, виникають етичні питання та відповідальність за можливі наслідки неправильних рішень.

Ефективність експертної системи залежить від доступності та співпраці з досвідченими експертами. Якщо немає можливості взаємодії з кваліфікованими фахівцями, це може обмежити потенційність системи.

Загалом, врахування цих обмежень важливо для успішного впровадження та використання експертних систем у конкретному контексті.

## 2 ЕКСПЕРТНІ СИСТЕМИ В ЗАДАЧ ЕЛЕКТРОЕНЕРГЕТИКИ

Головною метою створення експертних систем є полегшення роботи фахівців та автоматизація прийняття рішень. Здебільшого зараз використовують перетворення опису задачі та формалізації проблеми за допомогою природньої мови. Такий метод є необхідним через те, що є потреба у більш гнучких методах, які б обробляли інформацію, враховуючи невизначеність.

Фундаментом архітектури експертних систем є база знань щодо певної предметної області. Така база знань буде доповнюватися в процесі використання експертної системи. Головною перевагою експертних систем є можливість структуризації знань та їх представлення у зручному та зрозумілому вигляді для користувача.

Варто враховувати ще наявність різноманіття отриманих знань експертною системою, а саме, знання можуть бути чіткі, досить чіткі та нечіткі. Обробка та представлення таких знань виконується за допомогою логіки, адже може виникнути ситуація, для вирішення якої може бути недостатньо отриманої інформації або ж вона може бути повністю відсутня, тому тоді система повинна базуючись на своїх накопичених знаннях прийняти найбільш раціональне рішення для вирішення поставленої задачі. Такого виду рішення несуть ймовірнісний характер, а отже в таких ситуаціях необхідна теорія ймовірностей.

Враховуючи факт звернення до теорії ймовірності, можна сказати, що експертні системи використовують безпосередньо теорію Баєса для управління можливостями невизначеності.

$$P(A/B) = \frac{P(A/B) * P(A)}{P(B/A) * P(A) + P(B/\bar{A}) * P(\bar{A})}, \quad (1.1)$$

де А – деяка подія;

$B$  – інша подія;

$P(B/A)$  – умовна ймовірність виникнення події  $B$  за умови, що деяка подія  $A$  вже відбулася.

Таким чином ми отримуємо умовну ймовірність події  $B$  за умови  $A$ . Це надає нам можливість управління за умови невизначеності. З метою зменшення кількості умовних ймовірностей, які мали б бути визначеними, треба встановити зв'язок між умовно незалежними змінними з метою спростити обчислення результатів запитів.

## 2.1 Етап прийняття рішення

Експертні системи створені з метою прийняття оптимального рішення для поставлених задач. Для того, щоб експертна система могла коректно прийняти рішення, необхідно чітко та максимально повно задати цілі та описати поставлені задачі, а також треба коректно сформулювати та встановити критерії, за якими будуть обиратися оптимальні рішення поставлених цілей та задач, спираючись на набір даних та знань, які доступні експертній системі.

Також експертна система повинна мати можливість пошуку альтернативних шляхів вирішення поставлених задач, а отже база знань її повинна мати відповідне наповнення, щоб мати можливість обрати альтернативні варіанти. З усіх існуючих можливих варіантів вирішення задачі, експертна система повинна запропонувати найефективніший варіант та надати список за пріоритетністю альтернативних варіантів.

Надалі експертна система повинна обробити інформацію, яка надійшла та містить опис задачі та обрати критерій чи критерії за якими вона буде обирати оптимальний шлях вирішення задачі.

Необхідно враховувати, що в життєвих ситуаціях не завжди може існувати найкраще рішення, адже фактори, які впливають на вибір варіанту рішення задачі не завжди можуть бути однозначними, а отже експертна

система повинна вміти прийняти рішення, спираючись на свій попередній досвід та вже отримані знання.

## 2.2 Створення формального визначення логіки оперативних перемикачів у системах електропостачання

Наразі більша частина роботи диспетчерів проводиться без автоматизації процесів, а це значно сповільнює швидкість реакції та вирішення проблеми, яка виникла. В нинішніх реаліях діяти треба швидко, щоб уникнути можливих аварій та інших наслідків. Ми можемо наглядно побачити як цей процес налаштовано на прикладі нашої країни під час війни минулої зими, а в деяких містах і зараз, наприклад у Харкові. Було створено графіки відключень світла, налаштовано альтернативні джерела постачання електроенергії такі як сонячні панелі та інше, але нажаль все це здебільшого проводилося вручну диспетчерами.

Розглянемо більш детально сам процес. Спочатку диспетчер отримує заявку щодо певної несправності через телефон. Диспетчер вносить в базу всі необхідні дані щодо самого об'єкту, вид несправності, коли була отримана заявка, скільки треба часу на ремонт несправності. Далі дана заявка відправляється на розгляд та отримується відповідь щодо прийняття заявки в роботу чи відхилення. Наразі змінюється статус заявки та вона отримує свій номер на виконання. Кожна заявка також несе інформацію щодо самого диспетчера, який її прийняв та причину відмови опціонально. Зачасту, диспетчер приймає рішення щодо обраного альтернативного шляху електропостачання.

Для того, щоб обрати максимально вигідний альтернативний шлях потрібно враховувати досить багато факторів, а саме: необхідна потужність, режими роботи, кількість підключених об'єктів до підстанції та багато іншого. Тобто, якщо є необхідність у тривалій роботі та потужність не менше кількох мегават, то буде обрана паротурбінна ТЕЦ, а у випадку

досить швидкої зміни навантаження, то вже буде обрано або дизельна електроцентраль, або двигуни швидкого запуску.

Для промисловості є потреба у використанні безперебійного живлення, це обумовлено наявністю деякими видами обладнання, які потребують постійну наявність напруги, адже вони не можуть без неї стояти довгий час, бо це може призвести до аварійної ситуації або дуже складного повторного запуску всього виробництва. В таких випадках використовують синхронні генератори, синхронні двигуни з  $\cos\varphi$  0,9 або конденсаторні батареї.

Отже, для коректної роботи експертних систем, треба реалізувати алгоритм прийняття рішень, базуючись на формалізації логіки нечітких даних, а саме правил, базуючись на яких буде виконуватись комутація та перемикання пристроїв.

Під час створення експертних систем для обраної предметної області наразі застосовують автоматизована система перемикання розподільних мереж. Такий спосіб характеризується рядом рис:

- автоматичність: пристрої автоматичного перемикання, такі як вимикачі автоматики, дозволяють автоматично відключати пошкоджені ділянки мережі та перемикати навантаження на інші лінії або джерела живлення без втручання оператора;
- швидкість реакції: автоматизований спосіб перемикання забезпечує швидку відповідь на виникнення проблем, таким чином, мінімізуючи час перерви в електропостачанні абонентів;
- безпека та надійність: цей метод дозволяє зберігати безпеку мережі та мінімізувати вплив дефектів на інші частини системи, а також забезпечує надійність електропостачання абонентів;
- підтримка резервних джерел живлення: автоматизований спосіб перемикання часто використовується для перемикання на резервні джерела живлення у разі відмови основного джерела;

– автоматизована діагностика та відновлення: після виправлення дефекту автоматизований спосіб перемикання може автоматично повернути систему до нормального режиму роботи.

### 2.3 Способи оптимізації напруги в розподільній мережі

Забезпечення стабільної напруги у розподільних мережах є важливим завданням для забезпечення надійного та ефективного функціонування системи електропостачання. Для оптимізації напруги у розподільних мережах використовуються різноманітні методи та підходи, що спрямовані на забезпечення оптимальної роботи системи.

Один з методів оптимізації напруги полягає у використанні автоматичних регуляторів напруги (AVR) та розподілених регуляторів напруги (DVR), які автоматично регулюють рівень напруги у мережі шляхом керування трансформаторами та реактивним навантаженням. Ці регулятори можуть працювати на основі алгоритмів, які враховують вхідні дані про навантаження та стан мережі для оптимізації рівня напруги в різних точках системи.

Крім того, використання автоматизованих систем керування та моніторингу, таких як системи SCADA (Supervisory Control and Data Acquisition), дозволяє операторам мережі в реальному часі контролювати та регулювати рівень напруги в розподільних мережах з метою оптимізації ефективності та забезпечення нормальної роботи системи.

Крім того, впровадження розумних технологій та алгоритмів управління, таких як системи дистанційного керування та автоматичного аналізу даних, може сприяти оптимізації напруги шляхом прогнозування навантаження та вжиття відповідних заходів управління.

Отже, оптимізація напруги в розподільних мережах залежить від використання різноманітних методів та технологій, які спрямовані на забезпечення стабільної та надійної роботи системи електропостачання.



## 3 РОЗРОБКА МОДЕЛІ ЕКСПЕРТНОЇ СИСТЕМИ

Спочатку треба розглянути вже існуючі інформаційні процеси для розробки належного алгоритму роботи майбутньої експертної системи. Також необхідно проаналізувати існуючі програмні забезпечення та виділити оптимальне та найкраще для реалізації експертної системи, враховуючи розроблений алгоритм роботи на попередньому етапі. Завершуючим етапом розробки експертної системи буде саме тестування, для якого треба буде створити максимально реалістичні умови.

### 3.1 База знань

База знань – це структурована колекція інформації, яка зберігається та використовується для рішення завдань, вирішення проблем або надання підтримки користувачам. Вона може бути створена для будь-якої області знань та організована за допомогою різних методів, таких як бази даних, сховища даних, системи управління знаннями та інші.

База знань може містити різноманітну інформацію, таку як текстові описи, статті, документацію, ілюстрації, відео, аудіо та інші формати даних. Вона дозволяє зберігати, організовувати та швидко отримувати доступ до знань, що допомагає вирішувати проблеми та приймати рішення на основі попередньої експертної інформації.

Бази знань використовуються в різних сферах, таких як бізнес, наука, медицина, освіта, інформаційні технології та інші. Вони є важливим інструментом для підтримки прийняття рішень, розв'язання проблем та забезпечення ефективної передачі знань в організаціях та спільнотах.

Предметна область – це конкретна сфера знань, тема або область, що вивчається або досліджується. Вона може включати в себе певні теми, концепції, проблеми, процеси або об'єкти, які вивчаються з певної перспективи чи досліджуються для досягнення певних цілей.

Предметна область може бути широкою або обмеженою, в залежності від глибини та обсягу знань, які вона охоплює. Наприклад, предметною областю може бути медицина, фінанси, інформаційні технології, мови програмування, екологія, психологія та інші.

Вивчення предметної області може включати в себе аналіз концепцій, вивчення основних принципів, вирішення проблем, дослідження та розробку нових методів та технологій, а також застосування знань для практичних цілей.

Предметна область є важливою основою для розвитку знань, розуміння складних понять та вирішення реальних проблем у відповідних галузях діяльності. Вона визначає контекст та обсяг знань, які необхідно засвоїти або дослідити для розвитку кращого розуміння та досягнення цілей у відповідній області.

Експертні системи використовуються для моделювання та розв'язання складних проблем в різних предметних областях, шляхом використання знань та експертних правил. Подання знань у таких системах відіграє ключову роль у забезпеченні їх ефективності та функціональності.

Існують різні способи подання знань в експертних системах:

- продукційні правила (rules-based systems): цей метод використовує набір «якщо-тоді» правил, де вказуються умови (якщо) та висновки (тоді), які система використовує для вирішення проблем. Цей підхід особливо ефективний для моделювання експертних знань та логіки вирішення проблем у формі правил;

- системи бази знань (knowledge-based systems): ці системи використовують базу знань, яка зберігається у вигляді фактів, правил, керованих словників, або графічних моделей. Цей підхід дозволяє більш гнучко використовувати знання та легше модифікувати експертні системи;

- експертні системи на основі керованих словників (knowledge-based systems with controlled vocabularies): цей метод використовує керовані словники або термінологію для представлення знань у структурованій

формі. Він дозволяє стандартизувати та узгоджувати термінологію в рамках системи, спрощуючи роботу з знаннями;

– експертні системи на основі графічних моделей (knowledge-based systems with graphical models): цей підхід використовує графічні моделі, такі як схеми або діаграми, для представлення знань. Він може бути корисним для візуалізації складних взаємозв'язків між знаннями та логіки вирішення проблем.

Ці способи подання знань в експертних системах дозволяють ефективно моделювати та використовувати експертні знання для розв'язання різноманітних завдань та проблем у відповідних предметних областях.

В експертних системах, розроблених для моделювання та розв'язання складних завдань у різних предметних областях, знання може бути подане різними способами залежно від природи проблеми та потреб користувача. До найбільш поширених способів подання знань в експертних системах відносяться:

– продукційні правила: цей підхід базується на формалізованому використанні «якщо-тоді» правил, які відображають експертний досвід та логіку вирішення проблем. Продукційні правила вказують умови (якщо) та дії (тоді), які слід виконати при виконанні цих умов;

– керовані словники: цей метод використовує структуровані терміни та поняття для подання знань. Керовані словники спрощують процес введення та розуміння знань, оскільки вони стандартизують термінологію в рамках системи;

– бази даних знань: цей підхід передбачає використання баз даних для зберігання та організації знань. Знання подається у вигляді структурованих даних, що дозволяє ефективно здійснювати пошук, оновлення та використання інформації;

– експертні системи на основі графічних моделей: використання графічних моделей, таких як схеми або діаграми, дозволяє візуалізувати та розуміти складні зв'язки між знаннями та проблемами;

– фрейми: фрейми – це структуровані об'єкти, що містять інформацію про об'єкти або ситуації у предметній області. Кожен фрейм може містити атрибути та відношення до інших фреймів, що дозволяє ефективно організувати та використовувати знання.

Ці способи подання знань в експертних системах допомагають ефективно моделювати та використовувати експертні знання для розв'язання різноманітних завдань та проблем у відповідних предметних областях.

Динамічні та статичні бази знань є важливими компонентами експертних систем, які використовуються для зберігання та організації знань у відповідних предметних областях.

Використання статичних та динамічних баз знань в експертних системах допомагає забезпечити ефективне та надійне зберігання, організацію та використання знань для рішення складних проблем у різних предметних областях.

Статична база знань відображає сталу колекцію фактів, правил та понять у момент часу її створення. Вона не піддається змінам без спеціальних втручань і зазвичай використовується для статичних аналізів та вирішення проблем у відповідних областях. Статичні бази знань можуть бути особливо корисними в ситуаціях, коли знання є сталою та незмінною, а також в ситуаціях, де швидкість доступу до знань є критичною.

Динамічна база знань, натомість, може змінюватися та оновлюватися з часом з врахуванням нової інформації, отриманої в процесі роботи експертної системи або зовнішніх джерел. Вона дозволяє адаптуватися до змін в навколишньому середовищі та відображати останні дані та знання у відповідній області. Динамічні бази знань широко використовуються в

ситуаціях, де знання може швидко змінюватися або де постійне оновлення інформації є важливим для ефективного функціонування системи.

### 3.2 Модель представлення знань

Модель представлення знань – це структурований підхід до організації та представлення інформації, який використовується для зберігання, обробки та використання знань в комп'ютерних системах або експертних системах. Мета моделі представлення знань полягає у тому, щоб забезпечити систематизацію, структурування та доступність знань для ефективного використання.

Основні складові моделі представлення знань включають:

- символні об'єкти: це базові одиниці знань, які можуть бути представлені у вигляді символів, термінів або концепцій. Символьні об'єкти можуть включати різноманітні поняття, факти, події, правила, алгоритми тощо;
- відношення між об'єктами: це зв'язки та зв'язки між різними символними об'єктами, які вказують на їхні взаємозв'язки, асоціації або залежності. Відношення можуть бути однонаправленими або двонаправленими, сильними або слабкими, а також можуть мати різний ступінь важливості;
- фрейми та об'єкти: фрейм – це структурований об'єкт, який містить інформацію про певні типи об'єктів у предметній області. Фрейми можуть містити атрибути, значення, методи та інші характеристики об'єкту, що розширює можливості їх представлення;
- логічні правила та алгоритми: це набір логічних висловлювань, умов та дій, які визначають способи обробки знань та прийняття рішень на їх основі. Логічні правила та алгоритми можуть бути використані для автоматизованого аналізу, інтерпретації та використання знань.

Ці складові разом складають модель представлення знань, яка дозволяє системам ефективно зберігати, організовувати та використовувати знання для вирішення різних завдань та проблем у відповідних предметних областях.

Розглянемо класифікацію моделей подання знань:

1) мережеві:

- фрейми;
- семантичні мережі;

2) модульні:

- продукційні моделі;
- формальні логічні моделі.

Модульні моделі зазвичай використовуються для представлення поверхневих знань таких як правила чи аксіоми предметної області, а мережеві в свою чергу для глибинних знань таких як зв'язки у семантичних мережах чи фреймах.

### 3.2.1 Фрейми

Фрейми – це концептуальна модель для представлення знань, яка дозволяє описати об'єкти або ситуації у вигляді структурованих блоків даних. Кожен фрейм представляє собою сукупність атрибутів та їхніх значень, що характеризують певний об'єкт або концепцію.

Основні елементи фрейма включають:

- ім'я фрейма: унікальне ім'я, що ідентифікує фрейм та відрізняє його від інших фреймів;
- слоти: слоти визначають атрибути або властивості фрейма. Кожен слот може мати своє ім'я та значення;
- значення слотів: це конкретні значення, які призначені для слотів. Значення можуть бути різного типу, такого як текст, числа, списки, вказівники на інші фрейми тощо;

- підфрейми: фрейми можуть містити в собі інші фрейми як значення слотів. Це дозволяє створювати ієрархічні структури даних;

- методи: методи визначають поведінку або функції, що виконуються для фрейма. Вони можуть бути використані для обробки даних або взаємодії з іншими фреймами.

Фрейми дозволяють організувати та структурувати знання у вигляді логічних блоків, що спрощує їхнє представлення та обробку в комп'ютерних системах. Вони часто використовуються в експертних системах, базах даних, штучних інтелектуальних додатках та інших областях комп'ютерної науки для опису об'єктів та їхніх взаємозв'язків.

В базі знань застосовуються такі поняття як фрейми-зразки та фрейми-екземпляри для відображення реальних фактичних ситуацій на основі даних, що надходять.

Також через фрейм можна подати такі поняття:

- фрейми-структури для позначення об'єктів та понять;
- фрейми-ролі;
- фрейми-сценарії;
- фрейми-ситуації.

Структуру фрейму можна представити як перелік властивостей. Розглянемо структуру фрейму: ім'я фрейму, назва, значення, джерело значення, приєднання до слоту.

Для фреймів-екземплярів можна виділити декілька способів отримання значень слотом:

- від фрейму-зразка (Default-значення);
- успадкування властивостей від фрейму у слоті АКО.

Слот «АКО» (або «ISA» – «ієрархічний аспект») в контексті фреймів визначає відношення між фреймами, де один фрейм є підтипом (підкласом) іншого. Це вказує на те, що об'єкт, що описується в першому фреймі, є підтипом об'єкта, який описується в другому фреймі.

Наприклад, якщо у нас є фрейм «Транспортний засіб» та фрейм «Автомобіль», ми можемо встановити, що «Автомобіль» є підтипом «Транспортного засобу», використовуючи слот «АКО». Це означає, що кожен автомобіль є транспортним засобом.

Використання слота «АКО» дозволяє створювати ієрархічні структури фреймів, що спрощує організацію та розуміння взаємозв'язків між різними типами об'єктів у системі:

- за формулою, яка зазначена у слоті;
- процедура приєднання;
- діалог з користувачем;
- база даних.

Вигідною властивістю фреймів є успадкування властивостей, яке запозичене з семантичних мереж, яке відбувається за АКО-зв'язками. Неявне успадкування значень аналогічних слотів відбувається за допомогою слоту АКО, саме який і вказує на фрейм вищої ієрархії.

Фреймова модель представлення знань відображає концептуальну основу організації пам'яті людини та її гнучкість.

Розглянемо простий приклад фреймової моделі представлення знань для опису автомобілів.

Фрейм: Транспортний Засіб:

- слот: Назва;
- слот: Максимальна Швидкість;
- слот: Вага;
- слот: Тип Пального.

Фрейм: Автомобіль.

АКО Транспортний Засіб:

- слот: Кількість Дверей;
- слот: Потужність Двигуна;
- слот: Рік Випуску.

Фрейм: Вантажівка.



АКО Транспортний Засіб:

- слот: Максимальне Навантаження;
- слот: Довжина Кузова;
- слот: Кількість Вісей.

У цьому прикладі є три типи фреймів: «ТранспортнийЗасіб», «Автомобіль» та «Вантажівка».

Кожен фрейм має свої власні атрибути (слоти). Наприклад, фрейм «ТранспортнийЗасіб» має слоти для назви, максимальної швидкості, ваги та типу пального.

Фрейм «Автомобіль» є підтипом фрейма «Транспортний Засіб» (позначається використанням слоту «АКО») та має додаткові слоти, такі як кількість дверей, потужність двигуна та рік випуску.

Так само, фрейм «Вантажівка» також є підтипом фрейма «ТранспортнийЗасіб» і має свої власні атрибути, які характеризують вантажівки, такі як максимальне навантаження, довжина кузова та кількість вісей.

Ця фреймова модель представлення знань дозволяє описувати різні типи транспортних засобів та їх характеристики з урахуванням їхніх взаємозв'язків.

### 3.2.2 Семантична мережа

Семантична мережа – це графічна структура, яка використовується для представлення знань та їх взаємозв'язків у вигляді вузлів (або вузлів) та зв'язків між ними. Кожен вузол представляє собою концепцію або об'єкт, а зв'язки вказують на взаємозв'язки або відносини між цими концепціями.

Основні компоненти семантичної мережі включають:

- вузли (вершини, поняття): представляють концепції, об'єкти або сутності, про які міститься інформація в мережі. Наприклад, якщо ми

моделюємо семантичну мережу для родинних зв'язків, вузлами можуть бути «батько», «мати», «син», «донька» тощо;

– відношення (дуги): вказують на взаємозв'язки між вузлами і показують, які взаємодії або відносини існують між концепціями. Наприклад, зв'язок між вузлами «батько» і «син» може мати напрямок від «батька» до «сина», щоб показати батьківський зв'язок. Відношення ми формалізуємо за допомогою зв'язків типу «це», «належить», «має частину»;

– атрибути: деякі семантичні мережі можуть також містити атрибути, які характеризують вузли або зв'язки більш детально. Наприклад, для вузла «людина» атрибутами можуть бути «ім'я», «вік», «стать» і т. д.

Семантичні мережі застосовуються в різних областях, таких як штучний інтелект, обробка природної мови, бази даних та інформаційні системи. Вони дозволяють організовувати та представляти знання у вигляді структурованої графічної форми, що полегшує розуміння взаємозв'язків та використання інформації.

Семантичні мережі можна класифікувати за типами відносин між поняттями, а саме:

- бінарні (відносини між двома об'єктами);
- n-арні (відносини, які зв'язують більше двох понять).

Також класифікують за кількістю типів відносин, а саме однорідні та неоднорідні.

Здебільшого зустрічаються наступні типи відносин:

- логічні зв'язки (і, не, або);
- тимчасові (протягом, пізніше та інші);
- просторові (під, над, за, далеко від та інші);
- кількісні (дорівнює, менше, більше та інші);
- функціональні зв'язки (дієслова «впливає», «виробляє» та інші);
- зв'язок типу «частина-ціле» (автомобіль включає кермо);
- об'єкт класу (троянда, плетиста);

- значення властивості (колір має значення – жовтий);
- елемент класу (троянда це квітка);
- атрибутивні зв'язки (мати властивість) (пам'ять має властивість – обсяг).

Мінімальний склад відносин у семантичній мережі наступний:

- значення властивості;
- атрибутивні зв'язки (мати властивість);
- АКО чи об'єкт класу.

При використанні подібної моделі виникають складності, які пов'язані з організацією процедури виведення на семантичній мережі, тому дана задача зводиться до пошуку фрагмента мережі, яка відповідає певній підмережі, що відображає потрібний запит до бази.

Семантичні мережі мають кілька переваг, які роблять їх корисними для представлення та організації знань:

- структуроване представлення знань: семантичні мережі надають можливість структурувати знання у вигляді графічної мережі, що полегшує їх розуміння та використання;

- візуальна репрезентація: графічний характер семантичних мереж дозволяє легко візуалізувати зв'язки між концепціями та об'єктами, що полегшує розуміння складних взаємозв'язків;

- пошук та навігація: семантичні мережі дозволяють швидко виконувати пошук і навігацію між зв'язаними концепціями, що допомагає знаходити потрібну інформацію;

- можливість додавання нових вузлів і зв'язків: семантичні мережі можуть легко розширюватися шляхом додавання нових вузлів або зв'язків, що дозволяє їм адаптуватися до нових даних та сценаріїв;

- інтеграція з іншими системами: семантичні мережі можуть бути інтегровані з іншими системами, такими як бази даних або інформаційні системи, для розширення їхніх функціональних можливостей;

– підтримка розуміння мови: семантичні мережі можуть допомагати в розумінні природної мови шляхом моделювання семантичних відносин між словами та концепціями.

Ці переваги роблять семантичні мережі корисним інструментом для організації та використання знань у різних областях, від штучного інтелекту до баз даних та інформаційних систем.

Хоча семантичні мережі мають свої переваги, вони також мають деякі недоліки, які можуть обмежувати їх ефективність та застосування. Деякі з них включають:

– підвищена складність: при великій кількості вузлів і зв'язків семантична мережа може стати дуже складною для аналізу та розуміння. Це може призвести до збільшення часу обробки даних і складнощів у виявленні патернів або тенденцій;

– проблеми з моделюванням взаємозв'язків: деякі взаємозв'язки між концепціями можуть бути складними для експресії у вигляді зв'язків в семантичній мережі. Наприклад, взаємозв'язки, які вимагають багатofакторного розгляду або динамічних змін, можуть бути важкими для моделювання;

– недостатня гнучкість: семантичні мережі можуть бути менш гнучкими у порівнянні з іншими моделями знань, такими як правилі бази. Вони можуть бути складні для зміни або адаптації до нових сценаріїв або даних;

– проблеми з підтримкою неоднорідних даних: семантичні мережі можуть виявлятися неефективними у випадках, коли дані неоднорідні або важко структуровані. Вони можуть не бути оптимальним рішенням для представлення даних з великим обсягом недостовірної, неструктурованої або текстової інформації;

– проблеми з інтерпретацією зв'язків: іноді зв'язки між вузлами можуть бути невизначеними або суб'єктивними, що може викликати проблеми з інтерпретацією та розумінням семантичної мережі.

Незважаючи на ці недоліки, семантичні мережі залишаються потужним інструментом для представлення та розуміння знань у різних областях, таких як штучний інтелект, обробка природної мови, бази даних та інформаційні системи.

### 3.3 Продукційна модель

Продукційна модель представлення знань - це підхід до організації та представлення знань, який базується на використанні правил або «продукцій». В цій моделі знання виражаються у вигляді логічних правил, які вказують на умови (або «антецеденти») та дії (або «дії»), які потрібно виконати, якщо умови виконуються.

Основні складові продукційної моделі включають:

- антецеденти (умови): це умови або критерії, які повинні бути виконані, щоб виконати дію. Вони можуть включати перевірку наявності певних фактів або умов, які потрібні для активації правила;

- консеквенти (дії): це дії або операції, які виконуються, якщо умови правила виконуються. Ці дії можуть включати додавання нових фактів, зміну стану системи, виклик інших правил тощо;

- контекст: продукційні системи можуть також враховувати контекст або поточний стан системи під час виконання правил. Це дозволяє враховувати попередні дії та умови під час визначення, які правила слід виконувати.

Продукційні моделі часто використовуються в експертних системах, системах інтелектуального аналізу даних та системах прийняття рішень для автоматизації процесів виведення та прийняття рішень. Вони дозволяють ефективно представляти та використовувати знання у вигляді логічних правил, що спрощує програмування та розуміння поведінки системи.

Подібна модель дозволяє представити знання у пропозиції типу «Якщо (умова/антецедент), то (дія)».

Висновок в свою чергу зазвичай в такій ситуації буває прямий (від даних до пошуку мети) або зворотний (від мети для її підтвердження до даних, тобто вихідних фактів, які зберігаються в певній базі таких фактів).

Перевагою продукційної моделі та власне причиною її частого використання можна назвати наочність, простота внесення змін та доповнень, висока модульність, легкість механізму логічного висновку.

Давайте розглянемо приклад простої продукційної моделі для експертної системи, яка визначає, чи може людина отримати водійські права на основі віку та наявності досвіду водіння.

Правило: якщо людина має більше 18 років і має досвід водіння більше 2 років, то вона може отримати водійські права.

IF age > 18 AND experience > 2 THEN can\_drive = True.

Правило: якщо людина має більше 18 років і не має досвіду водіння більше 2 років, то вона не може отримати водійські права.

IF age > 18 AND experience <= 2 THEN can\_drive = False.

Правило: якщо людина має 18 років або менше, то вона не може отримати водійські права.

IF age <= 18 THEN can\_drive = False.

У цьому прикладі ми маємо три правила, які визначають, чи може людина отримати водійські права на основі віку та досвіду водіння. Кожне правило має свої антецеденти (умови) та дії. Якщо умова виконується, виконуються відповідні дії. Таким чином, експертна система може автоматично визначати, чи може людина отримати водійські права на основі введених даних.

Математичне подання продукційної моделі може бути виражене за допомогою логічних виразів та правил інференції. Давайте розглянемо математичне представлення прикладу продукційної моделі з визначенням можливості отримання водійських прав на основі віку та досвіду водіння.

Позначимо:

–  $A$  – вік особи;

- $E$  – досвід водіння особи;
- $P$  – можливість отримання водійських прав.

Тоді математичне подання правил буде виглядати наступним чином.

Якщо людина має більше 18 років і має досвід водіння більше 2 років, то вона може отримати водійські права:  $P=(A>18)\wedge(E>2)$ .

Якщо людина має більше 18 років і не має досвіду водіння більше 2 років, то вона не може отримати водійські права:  $P=(A>18)\wedge(E\leq 2)$ .

Якщо людина має 18 років або менше, то вона не може отримати водійські права:  $P=(A\leq 18)$ .

Таким чином, математичне подання продукційної моделі відображає логічні вирази, що визначають умови, за яких виконується можливість отримання водійських прав, на основі введених даних про вік та досвід водіння особи.

### 3.3.1 Формальні логічні моделі

Формальні логічні моделі – це математичні або логічні структури, які використовуються для представлення та вивчення різних видів логіки та виразування знань. Вони дозволяють формалізувати логічні відносини між об'єктами, фактами та правилами відповідно до певних синтаксичних та семантичних правил.

Деякі з основних формальних логічних моделей включають:

- пропозиційна логіка: використовується для виразування логічних виразів, які можуть бути істинними або хибними, використовуючи логічні операції, такі як кон'юнкція, диз'юнкція та заперечення;
- перша порядкова логіка: розширює пропозиційну логіку, дозволяючи виразити квантори, що описують кількість об'єктів, які задовольняють певні умови;
- модальна логіка: використовується для виразування можливостей та необов'язковості різних стверджень;

- дедуктивна база знань: це формальна модель для представлення знань та правил, що використовуються в експертних системах та інших інтелектуальних системах;

- модель представлення знань у вигляді продукційних систем: використовується для опису правил інференції в експертних системах;

- теорія множин: використовується для вивчення властивостей та відносин між множинами об'єктів;

- теорія графів: використовується для вивчення взаємозв'язків між об'єктами в графічній формі.

Ці формальні логічні моделі допомагають розуміти та вивчати логічні відносини між об'єктами та виражувати їх у вигляді формальних структур, що дозволяє ефективно аналізувати та використовувати знання в різних областях.

Зазвичай в реальних умовах обчислення предикатів першого порядку в промислових експертних системах майже не застосовується, лише у науково-дослідних системах, адже така система потребує високих умов та обмежень щодо предметної області. Тому у промислових ж експертних системах використовуються різні її модифікації і розширення, зміст яких виходить за рамки необхідної системи.

### 3.4 Вибір СУБД для експертної системи

База даних – це організована колекція даних, яка зберігається та управляється комп'ютерною системою з метою забезпечення ефективного доступу, зберігання та організації інформації. Основні характеристики бази даних включають:

- структура даних: база даних містить структуровану інформацію, організовану у вигляді таблиць, файлів чи інших структурних об'єктів, які визначаються схемою бази даних;



- можливість зберігання та оновлення: база даних дозволяє зберігати та оновлювати дані відповідно до вимог користувачів та додаткових обмежень, встановлених базою даних;

- забезпечення цілісності даних: база даних гарантує цілісність даних, тобто забезпечує їхню точність, достовірність та консистентність, шляхом застосування відповідних обмежень та правил;

- можливість одночасного доступу: база даних дозволяє одночасний доступ до даних від різних користувачів чи програмних засобів, забезпечуючи при цьому конкурентну обробку запитів;

- мови запитів та маніпуляції даними: база даних надає мови запитів та маніпуляції даними, які дозволяють користувачам отримувати доступ до даних та виконувати різноманітні операції з ними;

- безпека даних: база даних забезпечує захист даних від несанкціонованого доступу, використовуючи різноманітні методи шифрування, автентифікації та авторизації.

Бази даних використовуються в різних областях, таких як бізнес, наука, освіта, організація даних у великих веб-системах, ігрові індустрії та багато інших. Вони є невід'ємною складовою сучасних інформаційних систем та забезпечують ефективне управління та обробку даних. Існує ступінь розподіленості баз даних, а саме централізовані та розподілені.

Централізована база даних – це база даних, яка зберігається на одному центральному сервері та керується та обслуговується з центрального місця. У цій архітектурі всі дані зберігаються в одному місці, і всі операції з даними (зчитування, запис, оновлення тощо) виконуються на центральному сервері.

Основні характеристики централізованих баз даних:

- одна точка доступу: у централізованій базі даних всі дані доступні через один центральний сервер, що спрощує управління та керування даними;

- простота управління: централізовані бази даних можуть бути більш простими у встановленні, налаштуванні та управлінні, оскільки всі дані знаходяться в одному місці;

- загальний доступ до даних: всі користувачі отримують доступ до одного та того ж самого набору даних, що дозволяє забезпечити їхню консистентність та уніформність;

- зручність резервного копіювання: збереження резервних копій даних може бути спрощено, оскільки всі дані знаходяться в одному місці.

Однак централізовані бази даних мають деякі недоліки, такі як:

- точка відмови: якщо центральний сервер вийде з ладу, це може призвести до тимчасової недоступності всіх даних та послуг;

- обмежена масштабованість: обсяг та швидкість роботи централізованих баз даних можуть бути обмеженими через концентрацію всіх операцій на одному сервері;

- проблеми безпеки: якщо сервер скомпрометовано або атаковано, це може викликати серйозні проблеми з безпекою та конфіденційністю даних.

Централізовані бази даних зазвичай використовуються в невеликих та середніх підприємствах або у випадках, коли обсяг даних не є дуже великим і не вимагає великої масштабованості.

Розподілена база даних – це система зберігання даних, яка складається з ряду взаємопов'язаних баз даних, розташованих на різних комп'ютерах у мережі. Кожна база даних може бути фізично розташована на різних серверах у різних місцях, але вони функціонують як єдине логічне ціле для користувача та програм.

Основні характеристики розподілених баз даних:

- розподілена архітектура: дані розподіляються між кількома серверами, що дозволяє кожному серверу зберігати та обробляти певну частину даних;

- локальність даних: кожен сервер має локальний доступ до певної частини даних, що забезпечує швидкий доступ до даних без необхідності передачі їх через мережу;

- реплікація даних: деякі дані можуть бути розміщені на декількох серверах з метою підвищення доступності та надійності;

- масштабованість: розподілені бази даних забезпечують більшу масштабованість, оскільки можуть бути додані нові сервери для збільшення потужності та продуктивності системи;

- резервне копіювання та відновлення: розподілені бази даних можуть мати вбудовані механізми резервного копіювання та відновлення даних для забезпечення безпеки та надійності;

- глобальні операції: користувачі можуть виконувати операції з даними на різних серверах, що дозволяє виконувати глобальні операції над даними у розподіленій середовищі.

Розподілені бази даних часто використовуються у великих підприємствах та організаціях, де обсяг та обробка даних великі, а також у системах, що вимагають високої доступності та надійності. Вони дозволяють ефективно управляти та обробляти дані в розподіленому середовищі.

Розподілені бази даних викликають більше довіри, через те, що при відмові в роботі одного вузла мережі не зупиняється робота інших та існує резервна копія даних всіх вузлів у кожного вузла, тому такий вид баз даних і використовують у великих підприємствах та організаціях. Однак недоліком цієї системи можна назвати велику ресурсозатратність, що в межах експертної системи в якій обмежені ресурси як людські, так і матеріальні, то ми не можемо таку систему дозволити. Ще одним недоліком є невелика швидкість роботи розподілених баз даних, що викликано через роботу компонентів між собою через мережу. Такий недолік недопустимий для експертної системи, яка повинна в лічені секунди вирішити задачу щодо аварії на певній електромережі.

Отже, обгрунтувавши всі переваги та недоліки як централізованої БД, так і розподіленої, можна зробити висновки, що більш доцільно для нашої задачі буде використання саме централізованої БД. Щодо застосування такої БД, то є два варіанти: на сервері підстанції або на ЕОМ диспетчера. Також, як було зазначено раніше, що задля уникнення можливості втрати даних, необхідне використання та створення резервних копій. Задля забезпечення надійності збереження даних вирішено буде використовувати SSD-накопичувачі, адже вони більш надійні навідріз від жорстких дисків та мають більшу швидкість доступу до них.

Здебільшого для роботи з базами даних обирається мова SQL, адже вона стандартизована та зручна у використанні, але бувають деякі види задач, коли є потреба у більш широких можливостях мови. Для вирішення поставленої задачі в рамках дипломної роботи було обрано роботу з реляційними базами даних.

Реляційна база даних – це колекція даних, яка організована та зберігається відповідно до моделі даних, відомої як реляційна модель. У реляційній базі даних дані представлені у вигляді таблиць, які складаються зі стовпців та рядків. Кожна таблиця відображає певний тип даних, а кожен рядок представляє конкретний запис або кортеж, а кожен стовпчик – конкретну атрибуцію або поле даних. РБД підходить для великих підприємств, які мають фінансову можливість для впровадження передових технологій. Також РБД властива гнучкість структури даних, а саме зміна фізичної структури без зміни додатків.

Для виконання задачі в даній роботі було прийнято рішення використовувати СУБД ProtegeSQL.

ProtegeSQL – це інструмент, який поєднує в собі дві основні технології: інтелектуальні онтологічні моделі (за допомогою Protege) та бази даних (за допомогою SQL). Це розширення, яке дозволяє створювати та керувати базами даних, використовуючи онтологічні підходи.

Основні особливості ProtegeSQL включають:

- інтеграція онтологій із базами даних: завдяки ProtegeSQL можна створювати онтології та моделі знань, які потім можна інтегрувати з реляційними базами даних;

- мапування онтологій на реляційні структури: протегеSQL надає можливість встановлювати відповідності між онтологічними поняттями та структурами бази даних;

- запити до баз даних з використанням онтологічних понять: за допомогою ProtegeSQL можна виконувати SQL-запити до баз даних, використовуючи онтологічні терміни, що спрощує роботу з даними;

- підтримка мови SPARQL: ProtegeSQL підтримує мову запитів SPARQL для взаємодії з онтологічними даними, що розширює можливості роботи з знаннями.

СУБД ProtegeSQL підходить для вирішення більш складних задач, які потребують високої надійності та підтримки підходів об'єктних щодо баз даних, тому через низьку оптимізацію такої системи не доцільно використовувати для простих та більш дрібних задач.

Multi-Version Concurrency Control (MVCC) – це метод управління конкурентністю в базах даних, який дозволяє одночасно виконувати читання та записи даних без блокування таблиць або рядків. PostgreSQL використовує MVCC для забезпечення відмінної підтримки одночасного доступу до даних та транзакційної консистентності.

Основні характеристики MVCC в PostgreSQL:

- версіонування даних: кожне зміна в базі даних відображається як нова версія рядка. Старі версії рядків зберігаються разом з новими, що дозволяє транзакціям читати консистентні дані, які існують на момент їх початку;

- Read Committed Isolation Level: в режимі Read Committed транзакції бачать тільки ті дані, які були змінені до початку цієї транзакції. PostgreSQL використовує MVCC для забезпечення ізоляції транзакцій на рівні читання;

– без блокування для читачів: транзакції, які читають дані, не блокуються конфліктуючими записами, оскільки вони просто бачать попередні версії даних;

– блокування для записувачів: транзакції, які змінюють дані, блокуються тільки в тому випадку, якщо вони конфліктують з іншими транзакціями, які також змінюють ці дані;

– автоматичне видалення застарілих версій: PostgreSQL автоматично видаляє застарілі версії даних за допомогою механізму VACUUM, який підтримує консистентність та забезпечує ефективне використання простору в базі даних.

MVCC в PostgreSQL дозволяє забезпечити високий рівень одночасності та консистентності даних без зайвого блокування, що робить його дуже ефективним для великих та високонавантажених систем баз даних.

Ще однією перевагою для використання PostgreSQL є те, що ця система має можливість виконання процедур, які було написано та збережено на іншій мові, наприклад Java, Perl, Python, Ruby, Tcl, C та C++. Також існує безліч функцій в стандартній бібліотеці таких як математичні операції, криптографічні функції та багато іншого, що надає сумісність з Oracle.

### 3.5 Вибір програмного середовища розробки

Для вибору середовища програмної розробки треба враховувати такі аспекти як: експертна система повинна мати достатньо можливостей а також мати такі властивості: підтримка БД з метою накопичення певної інформації, що надає можливість у подальшому формувати ефективні та оперативні варіанти перемикань; кросплатформенність, що нам надає можливість звернення до експертної системи з будь-якого пристрою; об'єктно-орієнтована мова програмування, що нам надає можливість

створення якісного та зрозумілого візуалу, застосовуючи широкий спектр функціоналу; з метою підвищення рівня безпеки можливість зареєструвати користувача; налаштування прав доступу; облік тривалості робіт та частоти несправностей електричних мереж; оптимізація часу виконання алгоритму.

### 3.5.1 Середовище розробки IDE PhpStorm

PhpStorm – це інтегроване середовище розробки (Integrated Development Environment, IDE), призначене для роботи з PHP, HTML, CSS, JavaScript та іншими веб-технологіями. PhpStorm розроблений компанією JetBrains і відомий своєю потужністю, зручним інтерфейсом користувача та великим набором функцій для розробки веб-додатків.

Основні особливості PhpStorm включають:

- підтримка PHP: PhpStorm надає широкі можливості для розробки на мові програмування PHP, включаючи автоматичне завершення коду, перевірку синтаксису, рефакторинг, відлагодження та багато іншого;
- підтримка веб-технологій: PhpStorm також підтримує HTML, CSS, JavaScript, TypeScript, Sass, Less та інші веб-технології, надаючи інструменти для зручного редагування та відлагодження коду в цих мовах;
- універсальність: PhpStorm підтримує різні платформи, такі як Windows, macOS та Linux, що робить його доступним для широкого кола розробників;
- інтеграція з іншими інструментами: PhpStorm інтегрується з різними зовнішніми інструментами розробки, такими як системи контролю версій (наприклад, Git), фреймворки (наприклад, Laravel, Symfony) та інші;
- підтримка різних рамок розробки: PhpStorm надає підтримку для різних рамок розробки PHP, що дозволяє розробникам працювати з будь-яким фреймворком за своїм вибором;

– багатомовність: PhpStorm підтримує роботу з різними мовами програмування та технологіями, що дозволяє створювати комплексні веб-додатки з використанням різних інструментів та технологій.

PhpStorm є популярним інструментом серед веб-розробників завдяки своїй продуктивності, функціональності та зручності використання.

Головною перевагою обраного середовища розробки є підтримка SQL та баз даних, можливість розгорнути додатки віддалено, інтеграція з Git. PhpStorm також добре працює з Frameworks такими як: WordPress, Laravel? Magento, Yii, CakePhp.

Хоча PhpStorm вважається однією з найкращих IDE для розробки веб-додатків на мові PHP, вона також має свої недоліки (рисунок 3.1).

Розглянемо деякі недоліки:

– висока ціна: PhpStorm є комерційним продуктом, і його ліцензія може бути досить дорога для окремих розробників, студентів або малих підприємств;

– високі вимоги до ресурсів комп'ютера: PhpStorm вимагає відносно потужного комп'ютера з великою кількістю оперативної пам'яті та потужним процесором для ефективної роботи, що може бути проблемою для користувачів з обмеженими ресурсами;

– повільне завантаження: іноді PhpStorm може досить повільно завантажуватися, особливо на менш потужних комп'ютерах або при роботі з великими проектами;

– надмірне використання пам'яті: деякі користувачі відзначають, що PhpStorm може споживати багато пам'яті, особливо при відкритті багатьох файлів або роботі з великими проектами, що може призвести до зниження продуктивності комп'ютера;

– неінтуїтивний інтерфейс для новачків: для новачків інтерфейс PhpStorm може здаватися складним або неінтуїтивним, і їм може знадобитися деякий час, щоб освоїти всі його функції та можливості.



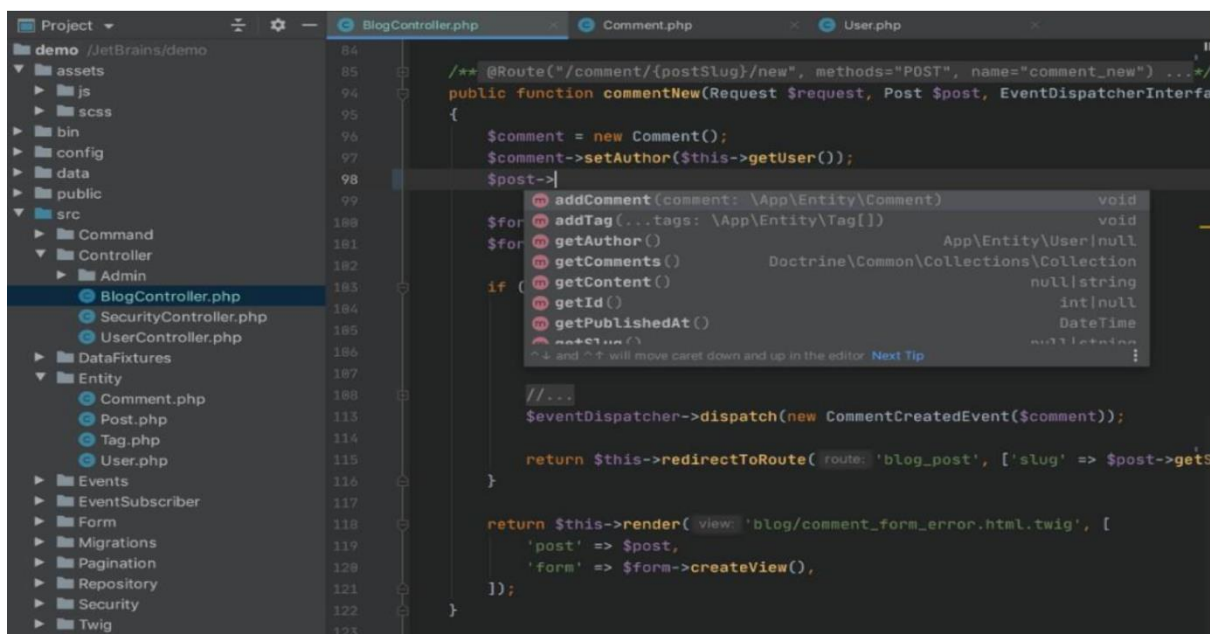


Рисунок 3.1 – Інтерфейс IDE PhpStorm

Хоча PhpStorm має свої недоліки, вона залишається дуже потужним та ефективним інструментом для розробки веб-додатків на мові PHP і користується популярністю серед професійних розробників.

### 3.5.2 Інтерфейс користувача

Для створення інтерфесу користувача, який би задовільняв всі потреби для експертної системи обраної предметної області, було прийнято рішення використовувати програмне забезпечення Unity 3D.

Unity 3D – це інтегроване середовище розробки програмного забезпечення, яке використовується для створення ігор, віртуальної реальності, доповненої реальності та інших інтерактивних додатків. Воно включає у себе різноманітні інструменти для розробки графічних ігор, а також підтримку різних платформ, таких як Windows, macOS, Android, iOS та інші.

Основні особливості Unity 3D включають:

- графічний двигун: Unity 3D має потужний графічний движок, який дозволяє створювати реалістичні графічні ефекти, анімацію та спеціальні ефекти;

- кросплатформеність: Unity підтримує різні платформи, що дозволяє розробникам створювати гри та додатки для різних пристроїв, включаючи ПК, консолі, мобільні телефони та інші;

- широкий вибір ресурсів: Unity має велику кількість безкоштовних та платних ресурсів, таких як моделі, текстури, анімації та звуки, які можуть бути використані для швидкої розробки ігор;

- скриптованість: Unity дозволяє розробникам використовувати скрипти на різних мовах програмування, таких як C#, JavaScript, Boo, щоб створювати власні функції та логіку гри;

- спільнота та підтримка: Unity має велику активну спільноту розробників та широку підтримку, що дозволяє знайти відповіді на будь-які питання та проблеми, які виникають під час розробки.

Unity 3D є одним з найпопулярніших інструментів для розробки ігор та інтерактивних додатків завдяки своїй потужності, гнучкості та широкому функціоналу.

Хоча Unity 3D є потужним інструментом для розробки ігор та інтерактивних додатків, він також має свої недоліки. Ось деякі з них:

- високі вимоги до апаратного забезпечення: робота в Unity може вимагати потужного комп'ютера з великою кількістю оперативної пам'яті та міцним процесором, щоб забезпечити плавну роботу середовища розробки та відтворення графіки;

- навчання та крива вивчення: для новачків може бути складно освоїти всі функції та можливості Unity через його розширений функціонал та складність деяких аспектів розробки;

- вартість ліцензій: хоча Unity має безкоштовний варіант, деякі функції та можливості доступні лише в комерційних версіях, які можуть бути вартістю для окремих розробників або невеликих студій.

– швидкість рендерингу: при створенні складних сцен з великою кількістю об'єктів та деталей може знижуватися швидкість рендерингу, що може призвести до проблем з продуктивністю та відтворенням графіки;

– залежність від платформи: перенесення гри з Unity на різні платформи може вимагати додаткового часу та зусиль, оскільки деякі функції можуть працювати по-різному на різних пристроях;

– великі розміри файлів: створення великих ігор або додатків у Unity може призвести до великих розмірів файлів, що може ускладнити завантаження та розповсюдження кінцевих продуктів.

Хоча Unity 3D має свої недоліки, він залишається одним з найпопулярніших інструментів для розробки ігор та інтерактивних додатків завдяки своїм багатофункціональним можливостям та широкій підтримці спільноти (рисунок 3.2).

Отже, обране програмне забезпечення повністю відповідає необхідним вимогам до ПЗ з метою створення якісної, зручної та інтуїтивно зрозумілої експертної системи.

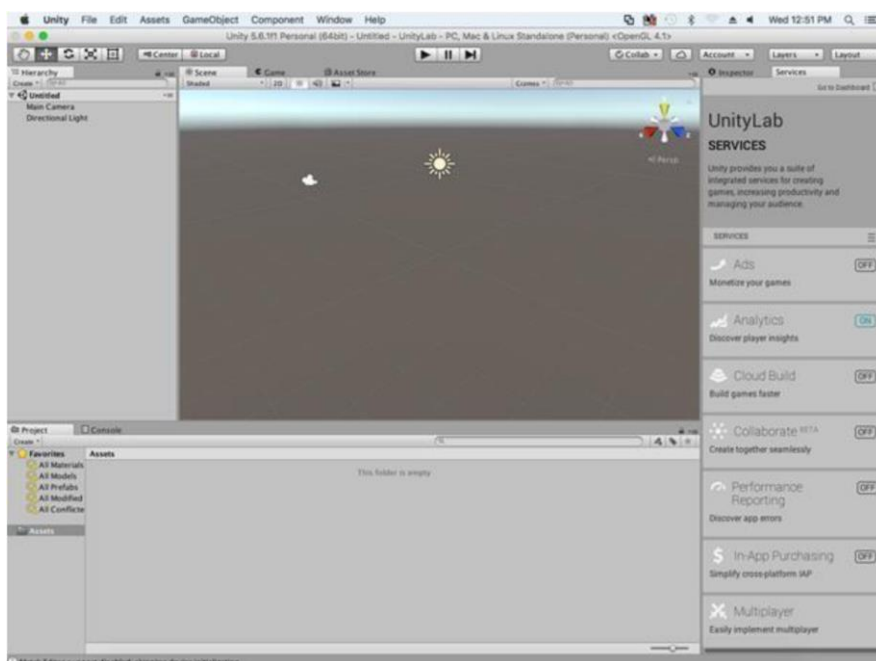


Рисунок 3.2 – Інтерфейс програмного забезпечення Unity 3D

### 3.6 Алгоритм обходу графа розподіленої електричної мережі

Від обраного алгоритму буде залежати саме швидкість роботи програми, а отже і швидкість отримання результату та налагодження несправностей.

Структура розподіленої електричної мережі виглядає як граф, а отже ми можемо звернутися до теорії графів з метою знаходження оптимального шляху. Існує метод пошуку у глибину або ширину, топологічне сортування та багато інших, але здебільшого алгоритми виконують обхід лише по вузлах найкоротшого шляху.

Або ж деякі алгоритми виконують надто багато заходжень у вузли з метою входження в кожен вузол та працюють за неорієнтованим графом. Також головною проблемою може стати складність обчислення деяких алгоритмів, адже вони прораховують всі існуючі маршрути до кожного вузла, а такий підхід в свою чергу заповільнить дуже роботу та збільшить час усунення аварії.

Потрібно розробити такий алгоритм, який буде відповідати критеріям для вирішення саме нашої задачі, а саме нам треба знаходження резервних джерел електропостачання за найкоротшим шляхом, щоб не було циклів та в пріоритеті були саме вихідні шляхи включення джерел.

Також має значення поняття вагів кожного джерела, щоб не було перенасичення шляху. Резервне джерело бажано підключати паралельно для надання надійності підключення; контроль уникнення підключення двох джерел одночасно, що може призвести до нової аварії; можливість враховувати вже збирані дані експертною системою з метою допомоги у подальших проектуваннях та розміщенні майбутніх підстанцій (рисунок 3.3).

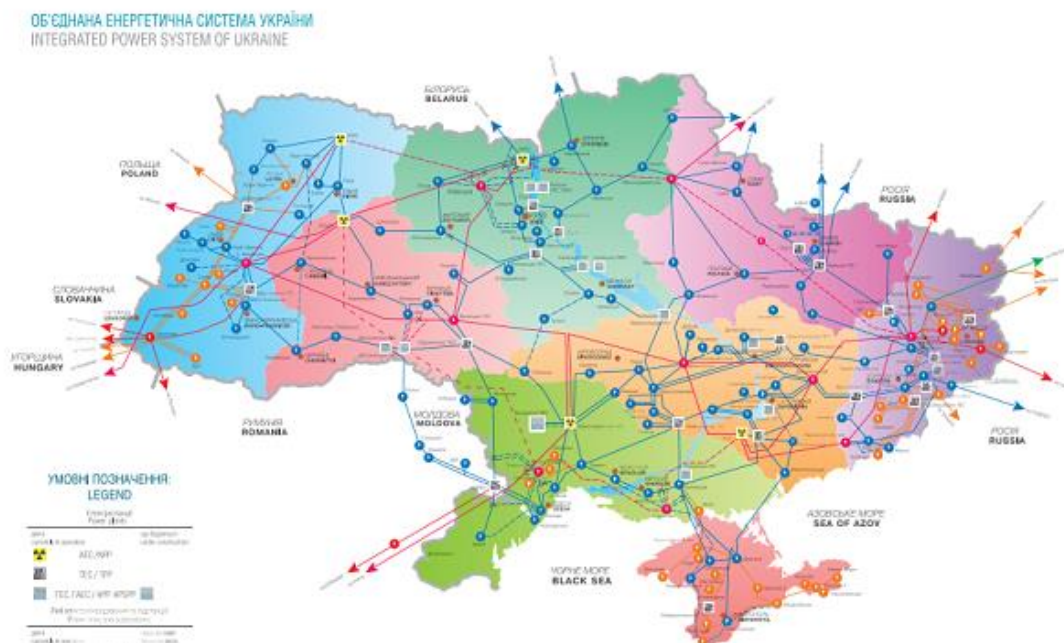


Рисунок 3.3 – Розподілена електрична мережа України

### 3.6.1 Алгоритм Дейкстри

Алгоритм Дейкстри – це алгоритм пошуку найкоротшого шляху в графі з невід'ємними вагами ребер.

Основна ідея алгоритму полягає в тому, щоб знайти найкоротший шлях від однієї вершини графа (зазвичай називається «початковою вершиною») до всіх інших вершин. Алгоритм позначає кожну вершину графа своїм відстанню від початкової вершини, а потім поступово оновлює ці значення, розглядаючи сусідні вершини та їх ваги. Процес продовжується до тих пір, поки не буде знайдено найкоротші шляхи до всіх вершин або досягнута цільова вершина, якщо така була вказана.

Основні кроки алгоритму Дейкстри:

- ініціалізація: встановлення відстані від початкової вершини до всіх інших вершин як нескінченність, крім початкової вершини, відстань до якої встановлюється як 0;

- вибір вершини: обирається вершина з найменшою відстанню, яку ще не відвідали;
- оновлення відстаней: для кожної сусідньої вершини, яка ще не відвідана, оновлюється відстань від початкової вершини, якщо новий шлях до неї коротший за поточний;
- повторення: процес обирання вершини та оновлення відстаней повторюється до тих пір, поки всі вершини не будуть відвідані або поки не буде досягнута цільова вершина;
- завершення: після завершення алгоритму отримуємо найкоротші шляхи від початкової вершини до всіх інших вершин графа.

Алгоритм Дейкстри ефективний лише в графах з невід'ємними вагами ребер і має складність  $O(V^2)$ , де  $V$  – кількість вершин у графі. Для реалізації більш ефективних алгоритмів пошуку найкоротшого шляху використовуються, наприклад, алгоритм Дейкстри з використанням черги з пріоритетами, який має складність  $O((V + E) * \log V)$ , де  $E$  – кількість ребер у графі.

### 3.6.2 Гамільтонів шлях

Гамільтонів шлях – це шлях у графі, який проходить через кожну вершину графа рівно один раз і повертається у початкову вершину. Іншими словами, Гамільтонів шлях – це шлях, який «відвідує» кожну вершину графа один раз без повторень і повертається до початкової вершини.

Одним із важливих застосувань гамільтонівих шляхів є задача комівояжера – пошук найкоротшого шляху, що проходить через кожне місто (вершину) рівно один раз і повертається до початкового міста (вершини). Вирішення цієї задачі важливе для оптимізації маршрутів доставки, логістики та інших практичних застосувань.

Для графа з  $n$  вершинами, гамільтонів шлях буде складатися з  $n$  вершин. Однак не в кожному графі існує гамільтонів шлях. В графах, в яких

існує гамільтонів шлях, вони можуть бути знайдені за допомогою різних алгоритмів пошуку, таких як алгоритми з використанням backtracking, dynamic programming, або простіші евристичні методи (рисунок 3.4).

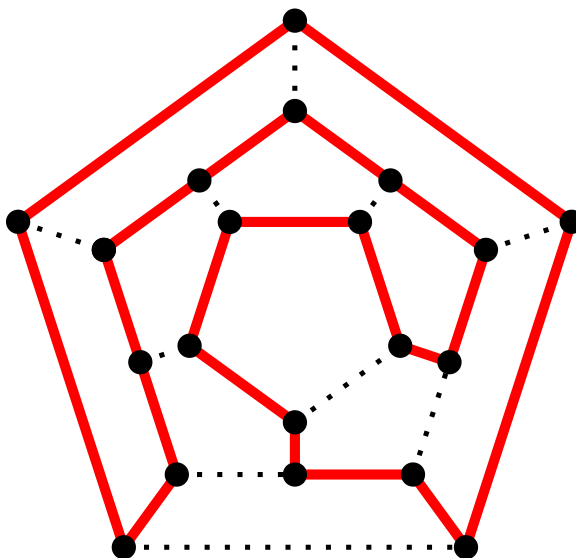


Рисунок 3.4 – Гамільтонів граф зі шляхом

Гамільтонів цикл – це випадок гамільтонівого шляху, в якому початкова вершина також є кінцевою. Тобто, це замкнутий шлях, який проходить через кожен вершину графа рівно один раз.

Гамільтонів граф – це граф, що містить гамільтонів цикл.

Граф, що містить простий ланцюг, що проходить через кожен його вершину, називається напівгамільтоновим.

Будемо вважати, що для кожної вершини  $X$  існує безліч вершин, які суміжні з кожною вершиною  $X$ . Будемо використовувати стек  $Path$  для збереження кожного етапу побудови шляху. Для кожної вершини  $X$ , яка збережена в стеку  $Path$ , зберігається множина всіх вершин  $N(x)$ , які суміжні з  $X$  та ще не приймали участь у розгляді в якості використання для подальшої побудови шляху з вершини  $X$ . Множина  $N(x)$  вважається рівною

тоді і тільки тоді, коли вершина  $X$  додана до шляху. Надалі розглянуті вершини видаляються з цієї множини. Далі ми повинні дослідити оточення останньої вершини  $X$  шляху  $Path$ . Якщо  $N(x) \neq 0$  та  $N(x)$  містить вершини, що не належать шляху, то одна з цих вершин додається до шляху, інакше ця вершина буде виключена зі стеку  $Path$ .

Далі необхідно перевірити чи всі вершини графу було задіяно у додаванні до шляху, якщо так, то треба перевірити чи суміжні між собою перша та остання вершини, якщо так, то ми отримали гамільтонів цикл (рисунок 3.5).

```
function findHamiltonianCycle(V,E):
  for v∈V:
    queue.pushBack(v)
  for k = 0..n * (n - 1)
    if (queue.at(0), queue.at(1)) ∈E
      i = 2
      while (queue.at(0), queue.at(i)) ∈E or (queue.at(1), queue.at(i + 1)) ∈E
        i++
      queue.swapSubQueue(1, i)
      queue.pushBack(queue.top())
      queue.pop()
```

Рисунок 3.5– Алгоритм проходження Гамільтонового шляху

В представленому прикладі алгоритму проходження гамільтонового шляху  $queue$  – це черга вершин графу  $G = \langle V, E \rangle$ .

Такий алгоритм є методом пошуку у глибину в межах дерева шляхів, а не графу. Вершини такого дерева – це прості шляхи, які мають початок з вершини  $A$ , тому ребра такого дерева – це з'єднання одного простого шляху, який виходить з іншого способом додавання в кінці ще однієї вершини.

Якщо розглядати даний алгоритм у якості використання для великих масштабів виробництва, то такий алгоритм не підходить, адже він працює лише з незваженими графами.



### 3.6.3 Вдосконалений алгоритм

Вдосконалити алгоритм можна шляхом поєднання алгоритму Дейкстри та Гамільтонового шляху.

Головною задачею при відновлення електропостачання на місцях аварії є зробити це якнайшвидше, а отже і маршрут необхідно буде обрати враховуючи мінімальну кількість перемикань та включення більшої кількості підстанцій.

Головні характеристиками, якими повинен володіти даний вдосконалений алгоритм є можливість виконувати роботи використовуючи зважені графи, виключення можливості виникнення ситуації підключення декількох резервних джерел до місця аварії.

## 4 РОЗРОБКА ТА РЕАЛІЗАЦІЯ АЛГОРИТМУ

Робота експертної системи загалом полягає саме в тому, щоб враховуючи вхідні дані провести їх аналіз та загалом всієї ситуації та спираючись на базу знань, попередні навички набуті системою, вивести висновки про стан місця аварії, яка мережа постраждала, які поруч є підстанції, які з них можуть стати потенційними резервними джерелами.

Такий аналіз експертної системи можна представити у вигляді дерева рішень, пройшовши яке ми прийдемо до результату, тобто до варіанту вирішення проблеми, яка виникла.

Результатом роботи ЕС буде процес порівняння (нормального і передбачуваного з'єднання), агрегування (схеми мережі), класифікація (аварія або резервний шлях), складання рапорту. Рапорт повинен відображати повний стан мережі і схему її комутації.

### 4.1 Розробка вдосконаленого алгоритму Дейкстри

Розроблений алгоритм повинен відповідати певним вимогам, завдяки яким і буде можливим його використання для розробленої експертної системи. До головних вимог до алгоритму можна віднести наступні властивості, а саме можливість працювати зі зваженими графами, застосування більшої кількості підстанцій, запобігання виникненню можливості підключення декількох резервних джерел.

Розглянемо вихідний стан електромережі у вигляді графа, де його ребра будуть мати такі атрибути:

- Id ребра;
- Id вузла;
- статус (активний/резервний/аварійний);
- зв'язані вузли;

- вага ребра (протяжність лінії електропередачі);
- атрибути вузлів графів;
- вага мінімального маршруту;
- Id вихідного вузла при розрахунку мінімального маршруту;
- статус вузла (активний/резервний/аварійний).

Зазначимо, що для вихідного стану електромережі, всі вузли знаходяться в активному стані. Було обрано кольори для позначення джерел, аварій та резервного маршруту. Зелений колір – джерела, червоний колір – аварія, жовтий – резервний маршрут (рисунок 4.1).

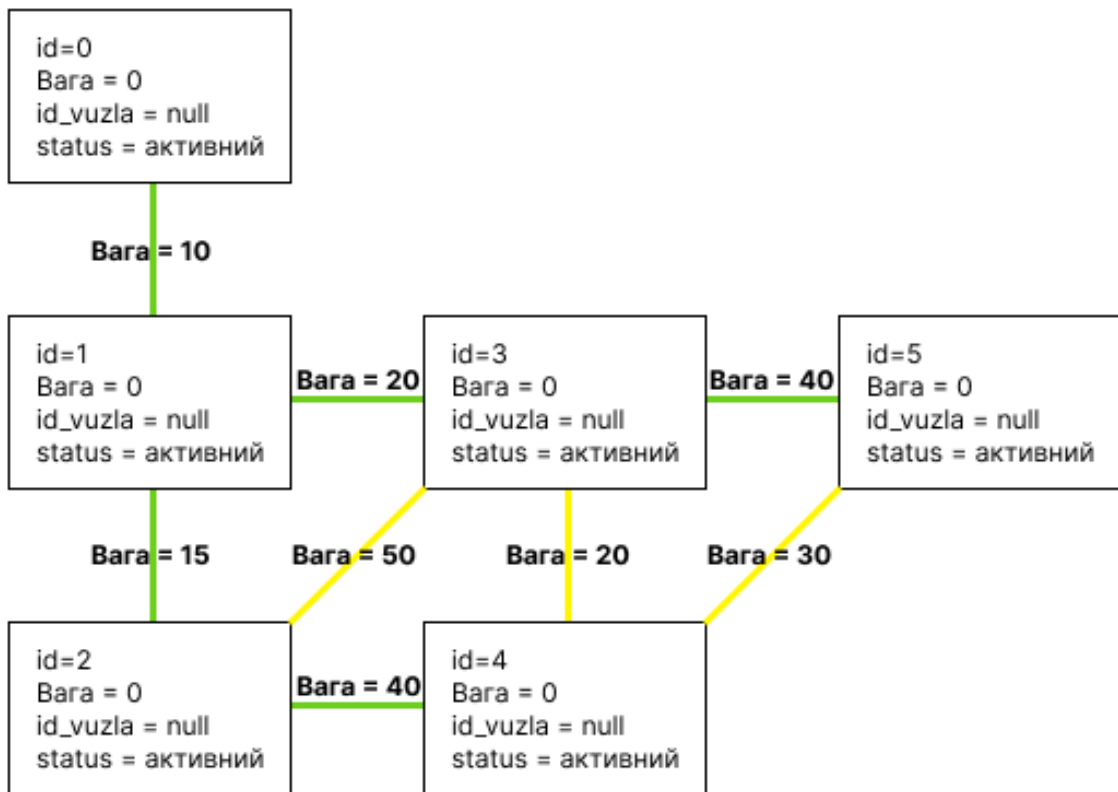


Рисунок 4.1 – Вихідний стан системи

Далі розглянемо на рисунках 4.2 та 4.3 покроково роботу розробленого алгоритму.

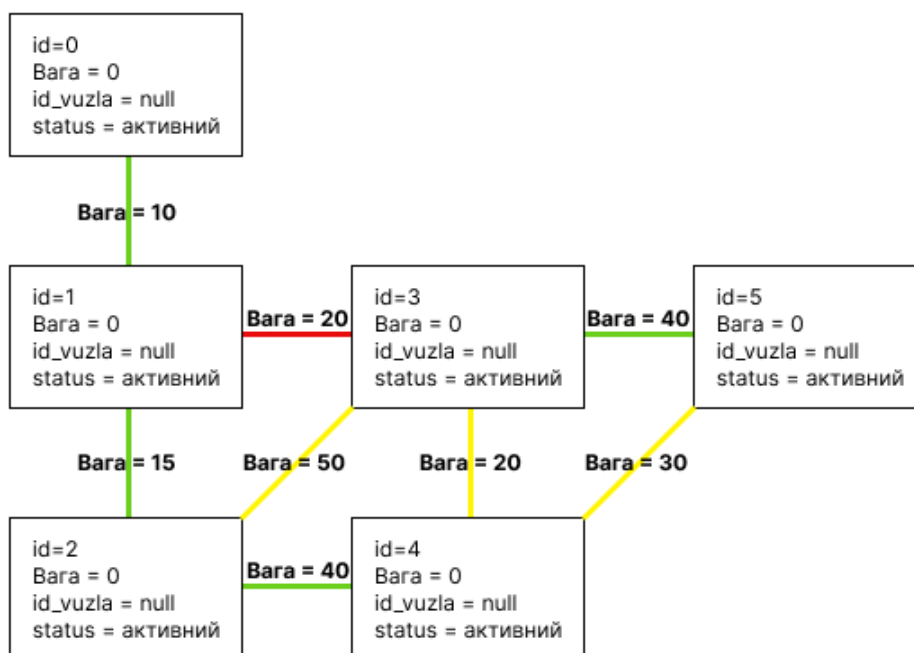


Рисунок 4.2 – Аварійний стан системи

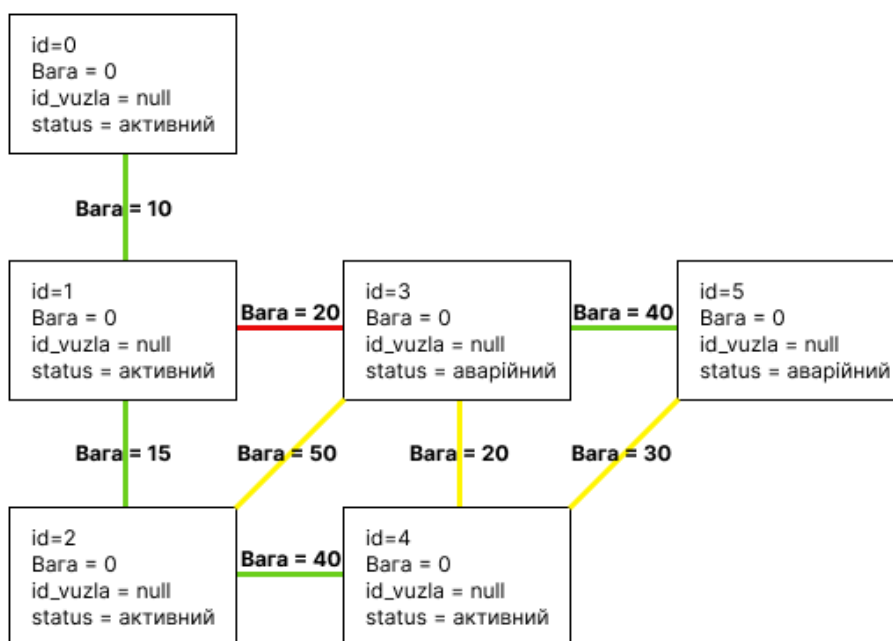


Рисунок 4.3 – Аварійний стан зв'язаних підстанцій

Наступним кроком за вдосконаленим алгоритмом буде поставити розрахункові ваги всіх вузлів в їхнє максимальне значення. Також іd вихідного вузла буде null під час розрахунку мінімального маршруту. Заповнюємо розрахункові вузли графу та іd вихідних вузлів щодо початкового вузла, тобто головної підстанції, використовуючи лише активні ребра заданого графу (рисунок 4.4).

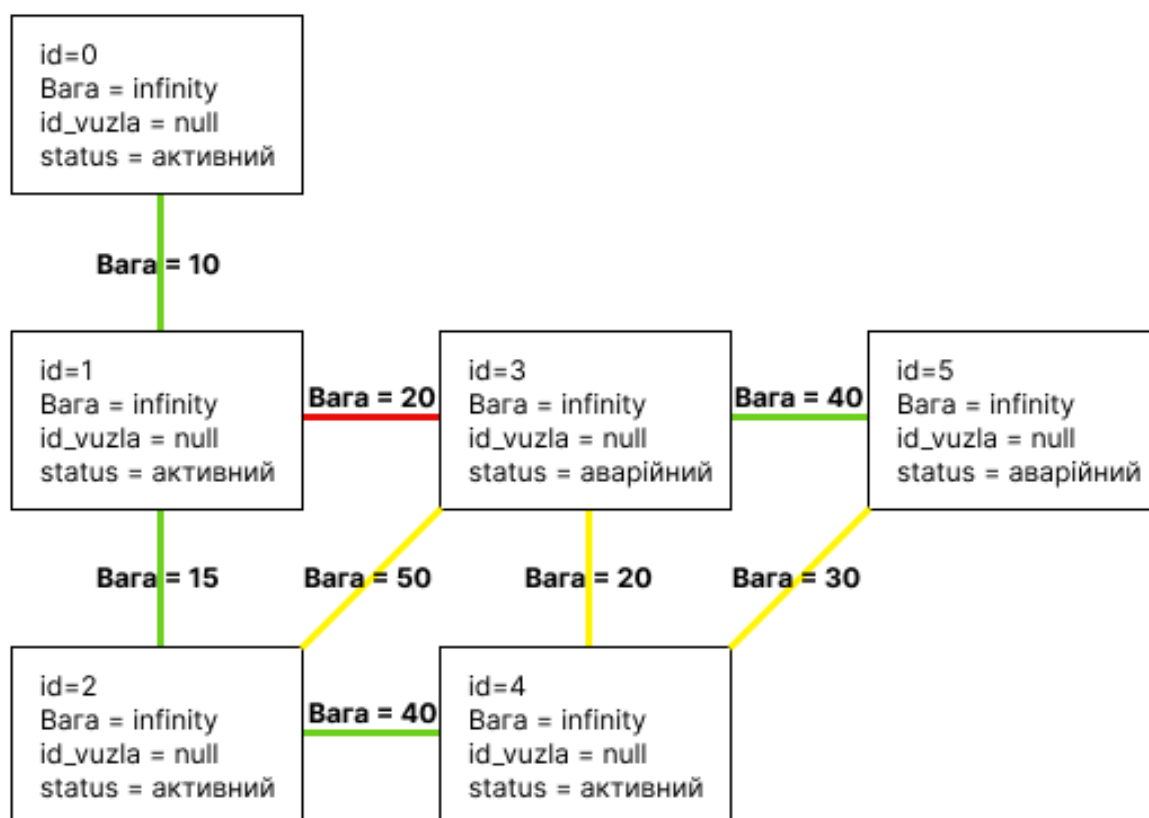


Рисунок 4.4 – Зміни у вагах та іd вихідних вузлів

Щоб розрахувати розрахункову вагу, необхідно знайти суму розрахункової ваги поточного вузла та ваги зв'язаного ребра. В той час іd вихідного вузла повинно бути рівним іd поточного вузла. Застосовуючи рекурсивне заповнення графу ми отримаємо граф, який відображає аварійний стан (рисунок 4.5).

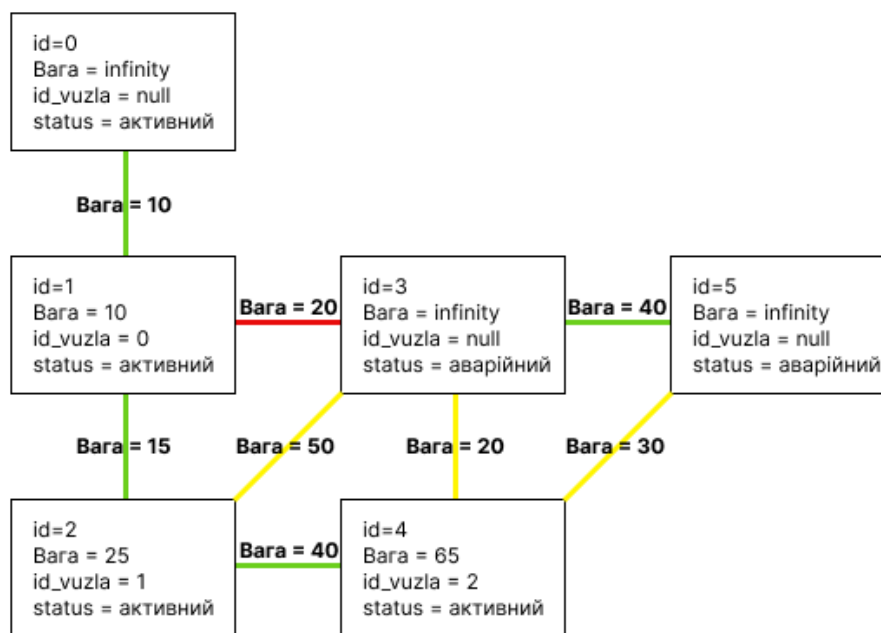


Рисунок 4.5 – Розрахунок вузлів

Наступним етапом алгоритму буде проходження по всім активним вузлам та заповнення графу надалі для резервних ребер.

Розрахункова вага обчислюється у випадку, коли зв'язаний вузол має стан «аварійний», звідси, якщо розрахункова вага зв'язаного вузла буде менша за суму розрахункової ваги поточного вузла та ваги зв'язаного ребра, тоді для зв'язаного вузла розрахункова вага буде обчислюватися наступним чином: Розрахункова вага = Розрахункова вага поточного вузла + вага зв'язаного ребра.

При заповненні графа ми встановлюємо іd вихідного вузла таким самим як і іd поточного вузла (рисунок 4.6).

Далі необхідно знайти мінімальне значення розрахункової ваги серед всіх вузлів зі станом «аварійний». Такий крок допоможе нам також знайти найкоротший шлях, на рисунку 4.5 можна побачити, що це вузол іd=3. Задля визначення тих ребер, які необхідно включити до найкоротшого шляху треба пройти по іd вихідних вузлів графа від вузла, знайденого в попередньому пункті, до досягнення активного вузла (рисунок 4.7).

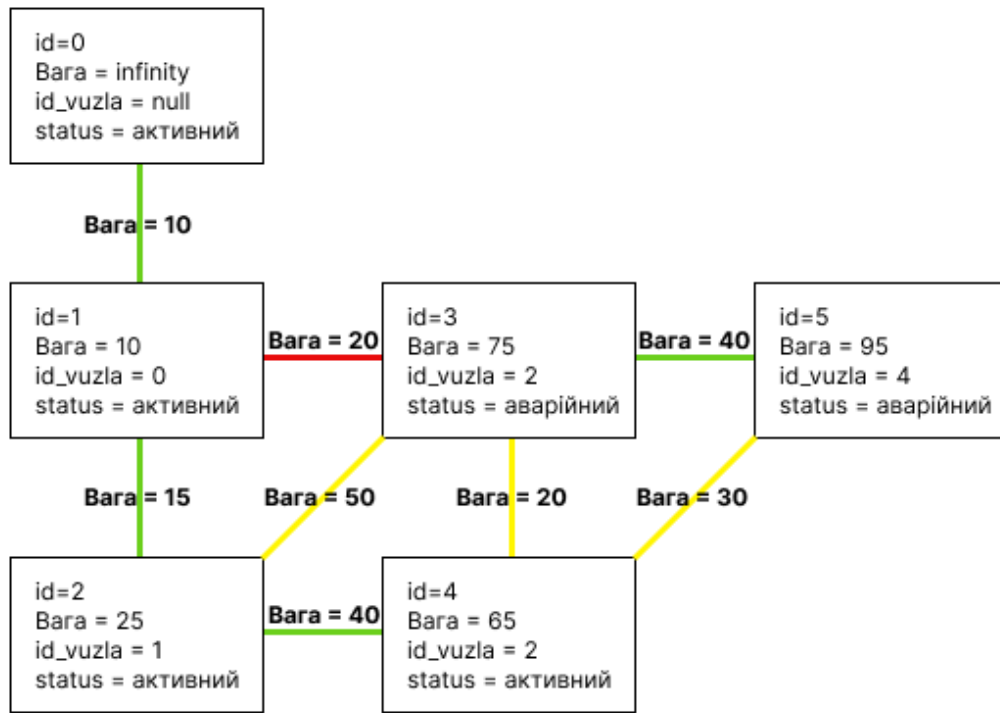


Рисунок 4.6 – Заповнення даних резервних ребер графу

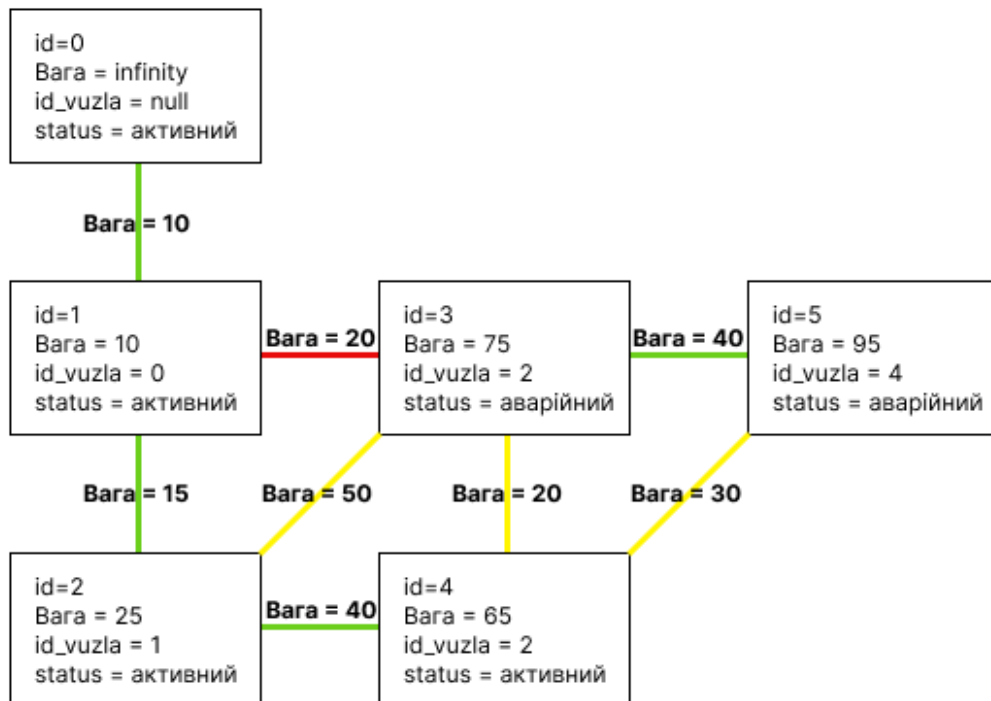


Рисунок 4.7 – Знайдений маршрут

Отже, як результат виконаної роботи, ми отримали розроблений алгоритм для вирішення поставленої задачі.

#### 4.2. Обрані технології для реалізації

Для реалізації розробленого алгоритму була обрана мова програмування Php із застосування фреймворку Laravel, для роботи з базою даних СУБД PostgreSQL і JS.

Обрані технології відповідають усім поставленим вимогам у попередніх розділах, а саме найголовніші з них це кросплатформеність та можливість підключення більше одного користувача. Також вони надають нам можливість реалізації у виробничих масштабах для виконання поставлених задач.

#### 4.3. Експертна система

Для реалізації експертної системи найголовнішим компонентом є саме база даних, яка реалізована застосовуючи SQL у вигляді 8 таблиць. Розглянемо деякі з них більш детально.

Таблиця «edges» (рисунок 4.8).

В цій таблиці представлено зв'язки, тому вона є однією з ключових, адже вона містить інформацію про ваги кожного джерела, порядковий номер та зв'язок між підстанціями:

Number – порядковий номер джерела;

Weight – вага джерела;

LinkedNode1\_id – початкова підстанція;

LinkedNode2\_id – кінцева підстанція;

Edge\_status\_id – статус джерела (активний, неактивний, аварійний, потребує включення).



id	number	weight	linkedNode1_id	linkedNode2_id	edge_status_id
1	1	3	1	2	1
2	2	5	2	3	3
3	3	4	3	4	1
4	4	5	4	5	3
5	5	2	4	6	3
6	6	2	6	7	1
15	15	3	1	5	1
24	24	20	2	4	2
52	52	4	5	2	2
53	53	8	3	5	1
56	56	620	5	6	4

Рисунок 4.8 – Таблиця «edges»

Таблиця «vertices» (рисунок 4.9).

В цій таблиці міститься інформація про статус вузла, положення, інформація щодо джерел електроживлення та інформація щодо попереднього вузла із ким зв'язаний.

id	number	weight	node_status_id	created_at	updated_at	x	y	primary	prev_node_id
1	1	(NULL)	1	2020-05-20 13:55:17	2020-06-05 16:23:36	0	-150	1	(NULL)
2	2	3	1	2020-05-20 13:55:17	2020-06-05 16:23:36	-200	-50	0	1
3	3	11	1	2020-05-20 13:55:17	2020-06-05 16:23:36	-200	150	0	5
4	4	15	1	2020-05-22 22:10:39	2020-06-05 16:23:36	100	150	0	3
5	5	3	1	2020-05-22 22:12:18	2020-06-05 16:23:36	100	-50	0	1
6	6	623	3	2020-05-23 14:56:40	2020-06-05 16:23:36	200	150	0	5
7	7	(NULL)	3	2020-05-23 14:57:46	2020-06-05 16:23:36	300	150	0	(NULL)

Рисунок 4.9 – Таблиця «vertices»

Weight – величина, яка демонструє пріоритет перемикань, який вираховано програмно.

Node\_status\_id – демонструє статус підстанції X-координата на X візуалізації Y-координата на Y візуалізації.

Primary – демонструє джерело електроживлення.

Prev\_node\_id – попередня підстанція (розраховується алгоритмом, на основі даних з таблиці «edges»).

За допомогою даної таблиці можна дуже розширити можливості експертної системи, наприклад додавання іншого джерела, опис підстанцій, адресація підстанцій до конкретного джерела живлення та інше.

Таблиці «vertex\_status» та «edge\_status» містять інформацію щодо статусу підстанцій та джерел, про кольори статусів (рисунок 4.10).

id	name	created_at	updated_at	color
1	Активен	2020-05-20 13:55:17	2020-05-20 13:55:17	green
2	Неактивен	2020-05-20 13:55:17	2020-05-20 13:55:17	orange
3	Аварийный	2020-05-20 13:55:17	2020-05-20 13:55:17	red
4	Необходимо включить	2020-05-22 14:38:00	2020-05-22 14:38:00	blue

Рисунок 4.10 – Таблиця «edge\_status»

Таблиці «user», «reset\_password» та «failed\_jobs» створені для розширення функціоналу експертної системи, а саме надає можливість додавати користувачів та надавати їм права доступу, дивитися час та дату відмов і час витрачений на ремонт та інше.

#### 4.4 Тестування

Розглянемо роботу експертної системи наглядно на графі. Зелений колір – робочий стан, жовтий – резервне живлення, червоний – аварія, синій колір – потребує ввімкнення (рисунок 4.11).

Якщо перевести джерела 5–4 в аварійний стан та натиснути кнопку «Пошук шляху», то отримаємо такий результат (рисунок 4.12).

Тепер ускладнимо завдання та додамо ще одну аварію на джерелі 3–4 (рисунок 4.13).

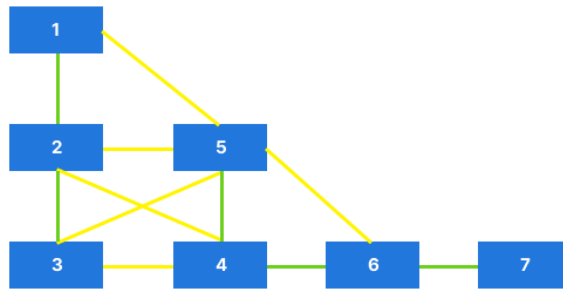


Рисунок 4.11 – Початковий стан графу

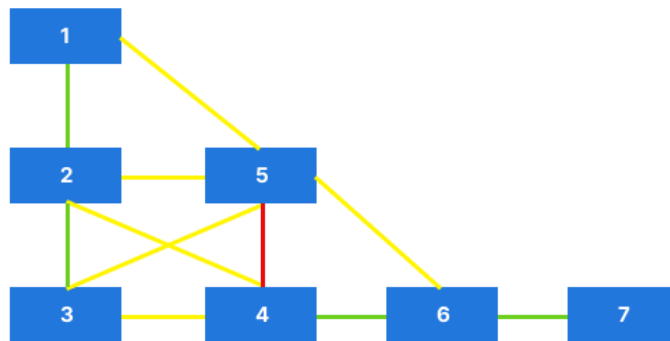


Рисунок 4.12 – Результат роботи за умови пошкодження джерела 5–4

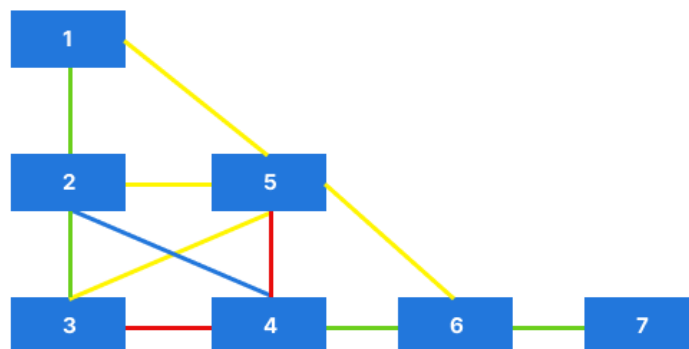


Рисунок 4.13 – Результат роботи за умови пошкодження джерела 5–4

та 3–4

Під час роботи експертна система наглядно змінює кольори на графі, а також змінює і самі значення схеми комутації задля використання змінених даних у подальшому аналізі (рисунок 4.14).

👉 id	number	weight	👉 linkedNode1_id	👉 linkedNode2_id	👉 edge_status_id
1	1	3	1	2	1
2	2	5	2	3	1
3	3	4	3	4	3
4	4	5	4	5	3
5	5	2	4	6	1
6	6	2	6	7	1
15	15	3	1	5	1
24	24	20	2	4	4
52	52	4	5	2	2
53	53	8	3	5	2
56	56	620	5	6	2

Рисунок 4.14 – Змінена структура баз даних

Отже, можна зробити висновки, що розроблений алгоритм працює вірно та досить ефективно за допомогою мінімальних перемикачів.

## ВИСНОВКИ

В даній кваліфікаційній роботі було досліджено експертні системи, які базуються на знаннях та за допомогою яких можна було б спростити та пришвидшити роботу диспетчера та підвищити якість результату роботи та вирішення задачі.

Було проведено аналіз вже існуючих інтелектуальних інформаційних систем для виконання задач діагностики та досліджено детально їх функціонал та інтерфейс.

Також було проведено детальний аналіз принципу роботи диспетчера та визначення необхідного функціоналу експертної системи для вирішення задач в області електропостачання.

В рамках кваліфікаційної роботи було визначено актуальність теми, область застосування, переваги такої системи та перелік задач, які може вирішити такого виду інтелектуальна інформаційна система.

Було проведено детальний аналіз та порівняння вже існуючих алгоритмів вирішення поставленої задачі та розроблено свій вдосконалений алгоритм Дейкстри, який дозволяє виконати поставлену задачу із мінімальною кількістю перемикачів та враховуючи найкоротший шлях.

В результаті можна зробити висновки, що впровадження розробленої експертної системи є необхідним та ефективним, адже така система можлива у застосуванні на великих підприємствах тому робота диспетчера значно полегшиться та стане більш продуктивною та ефективно як у часі виконання так і в результаті виконання.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

- 1) Recommender systems survey / J. Bobadilla et al. *Knowledge-Based systems*. 2013. Vol. 46. P. 109–132.  
URL: <https://doi.org/10.1016/j.knosys.2013.03.012> (date of access: 19.03.2024).
- 2) Филипов В. О. Інформаційні взаємодії та Web-сервіси. Ленанд, 2009. 144 с.
- 3) Громов Г. Ю. Введення в реляційні бази даних. 2008. 43 с.
- 4) Мюллер Р. Дж. Бази даних та UML. Проектування. Лори, 2002. 420 с.
- 5) Березніков, В.П. Організація обчислювального процесу при автоматизованій побудові діалогу в системі навчання операторів енергопідприємств. Київ, 1988. 20 с.
- 6) Філатов О.О. Перемикання в електричних розподільних пристроях. Москва: Енергія, 1985. 416 с.
- 7) Aiello, N.A. Comparative study of control strategies for expert systems: AGE implementation of three variations of PUFF / N.A. Aiello // In Proc. National Conference on Artificial Intelligence. 1986. С. 10–40.
- 8) Boose, J.H., Bradshaw, J.M. Expertise transfer and complex problems: using AQUINAS as a workbench for knowledge-based systems / J.H. Boose, J.M. Bradshaw // International Journal of Man-Machine Studies. 1987. Vol. 26. P. 1–28.
- 9) Мандрикін С.О., Філатов О.О. Експлуатація та ремонт електрообладнання станцій та мереж. Вища школа, 1983. 344 с.
- 10) Самойлов, В.Д., Березніков, В.П., Писаренко, А.П., Сметана, С.І. Автоматизація побудови тренажерів і навчальних систем. Київ: Наукова Думка, 1989. 197 с.
- 11) Орнов В.Г., Рабінович М.О. Завдання оперативного і автоматизованого управління енергосистемами. Вища школа, 1988. 221 с.

- 12) Boose, J.H., Gaines, B. Knowledge Acquisition Tools for Expert Systems. New York: Academic Press, 1988. С. 7–65.
- 13) Buchanan, B.G., Shortliffe, E.H. Rule-Based Expert Systems. Reading, MA: Addison-Wesley, 1984. С. 10–40.
- 14) Lipmann R.P. Pattern classification using neural networks. IEEE Communications Magazine, 1989. С. 47–67.
- 15) Naylor C. Build your own expert system. Chichester: John Wiley&Sons Ltd, 1987. 289 с.
- 16) Бодяньський, Є.В., Руденко, О.Г. Штучні нейронні мережі: архітектура, навчання. Харків: ТЕЛТЕХ, 2004. 372 с.
- 17) Купершмідт, Ю.Я., Любарський, Ю.Я., Орнов, В.Г. Принципи побудови універсального програмованого тренажера оперативних перемикачів // Електричні станції. 1982. С. 48–52
- 18) Суботін, С.О. Подання та обробка знань у системах штучного інтелекту та підтримки прийняття рішень: навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.
- 19) Стефік, М., Ейкінс, Я., Балзер, Р. Організація експертних систем // Кібернетичний збірник. Видання 22. Москва: Мир, 1985. С. 170–221.
- 20) Івашко, В.Г., Фінн, В.К. Експертні системи і деякі проблеми їх інтелектуалізації // Смуток і інформатика. Видання 27. Москва: ВІНІТІ, 1986. С. 25–61.
- 21) Боуман, Д., Емерсон, С.Л., Дарновські, М. Практичний посібник SQL. Використання мови структурованих запитів. Москва: Вільямс, 2001. 336 с.
- 22) PostgreSQL. URL: <https://www.postgresql.org/docs> (дата звернення: 09.04.2024).
- 23) Borabai. Фізичне проектування бази даних. URL: <https://bourabai.ru/dbt/dbms/03.html> (дата звернення: 21.04.2024).
- 24) ДСТУ 3008:2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Київ: ДП «УкрНДНЦ», 2016. 31 с.

25) Загальні методичні вказівки з дипломного проектування в університеті. Харків: ХНУРЕ, 2003. 40 с.

26) Методичні вказівки відносно структури, змісту та оформлення навчально-методичної літератури для авторів (упорядників) оригіналів. Харків: ХНУРЕ, 2002. 60 с.

27) Семенець В В, Філатов В О. Інтегровані технології в галузі інтелектуальних обчислень // Інтелектуальні системи автоматизації: монографія / Аврунін О Г та ін. – Кременчук: Новабук, 2021. – Розд. 5. – С. 275-297.

28) Filatov V, Semenets V, Zolotukhin O. Data mining in relational systems // *Innovative Technologies and Scientific Solutions for Industries*. 2020. №3 (13). С. 65–76. DOI: 10.30837/ITSSI.2020.13.065.