

Державний торговельно-економічний університет
Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Експертна система з байєсівською системою логічного
висновку»**

Студента 4 курсу, 9 групи,
спеціальності
122 «Комп'ютерні науки»

підпис студента

Давидовський
Владислав
Володимирович

Науковий керівник
кандидат технічних наук, доцент

підпис керівника

Демідов Павло
Георгійович

Гарант освітньої програми
кандидат технічних наук, доцент

підпис керівника

Демідов Павло
Георгійович

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Пурський О.І.
«20» грудня 2022 р.

Завдання на випускню кваліфікаційну роботу (проект) студенту

Давидовському Владиславу Володимировичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту):

«Експертна система з байєсівською системою логічного висновку»

Затверджена наказом ректора від «09» грудня 2022 р. № 3332

2. Строк здачі студентом закінченої роботи «31» травня 2023 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: обґрунтування та розробка експертної системи з байєсівською системою логічного висновку (на прикладі системи охоронної тривожної сигналізації).

Об'єкт дослідження: процеси функціонування системи охоронної тривожної сигналізації (ОТС).

Предмет дослідження: моделі, методи та алгоритми створення експертної системи охоронною тривожної сигналізації з використанням байєсівських мереж.

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Демідов П.Г.	24.12.2022	24.12.2022
2	Демідов П.Г.	24.12.2022	24.12.2022
3	Демідов П.Г.	24.12.2022	24.12.2022

6. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ТА АНАЛІЗ СТАНУ РОЗРОБОК ЕКСПЕРТНИХ СИСТЕМ РОЗВ'ЯЗАННЯ ІНТЕЛЕКТУАЛЬНИХ ЗАДАЧ

1.1. Поняття, призначення та структура експертної системи

1.2. Моделі представлення знань та їх порівняльна характеристика

1.3. Мови та технології програмування штучного інтелекту

РОЗДІЛ 2. ПРОЕКТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ З БАЙЄСІВСЬКОЮ СИСТЕМОЮ ЛОГІЧНОГО ВИСНОВКУ

2.1. Постановка задачі на розробку експертної системи охоронної тривожної сигналізації на базі байєсівської мережі довіри

2.2. Математична модель міркування в байєсівських мережах

2.3. Алгоритм розрахунку імовірнісних параметрів експертної системи охоронної тривожної сигналізації

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЕКСПЕРТНОЇ СИСТЕМИ З БАЙЄСІВСЬКОЮ СИСТЕМОЮ ЛОГІЧНОГО ВИСНОВКУ

3.1. Програмна реалізація експертної системи ОТС

3.2. Представлення байєсівської мережі довіри експертної системи ОТС та вхідних її імовірнісних параметрів в пам'яті комп'ютера

3.3. Розробка, налагодження та виконання програмного коду експертної системи ОТС в середовищі пакету SWI-Prolog

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	<i>01.10.2022</i>	<i>03.10.2022</i>
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	<i>15.12.2022</i>	<i>15.12.2022</i>
3	<i>Вступ</i>	<i>03.02.2023</i>	<i>03.02.2023</i>
4	<i>РОЗДІЛ 1. Теоретичні відомості та аналіз стану розробок експертних систем розв'язання інтелектуальних задач</i>	<i>28.02.2023</i>	<i>28.02.2023</i>
5	<i>РОЗДІЛ 2. Проектування експертної системи з байєсівською системою логічного висновку</i>	<i>06.04.2023</i>	<i>06.04.2023</i>
6	<i>РОЗДІЛ 3. Розробка програмного забезпечення експертної системи з байєсівською системою логічного висновку</i>	<i>12.05.2023</i>	<i>12.05.2023</i>
7	<i>Висновки</i>	<i>16.05.2023</i>	<i>16.05.2023</i>
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	<i>15.05.2023</i>	<i>15.05.2023</i>
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	<i>26.05.2023</i>	<i>26.05.2023</i>
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	<i>27.05.2023</i>	<i>29.05.2023</i>
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	<i>30.05.2023</i>	<i>30.05.2023</i>
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	<i>За розкладом роботи ЕК</i>	

8. Дата видачі завдання «24» грудня 2022 р.

Анотація

У даній випускній кваліфікаційній роботі проведено дослідження та розробку експертної системи з використанням байєсівської системи логічного висновку. Робота має на меті вирішення проблеми невизначеності в експертних системах та покращення процесу прийняття рішень.

Аналізуються теоретичні основи байєсівського висновку та його застосування в експертних системах. Розробляється архітектура експертної системи, включаючи базу знань, систему логічного висновку та інтерфейс користувача. Застосування байєсівської системи логічного висновку досліджується на конкретних прикладах і випадках використання. Розробляється програмне забезпечення для реалізації експертної системи та проведення випробувань. В результаті отримано експертну систему з байєсівською системою логічного висновку, яка демонструє здатність до ефективного вирішення завдань прийняття рішень в умовах невизначеності.

Ключові слова: експертна система, байєсівська система, логічне висновування, невизначеність, прийняття рішень.

Annotation

In this final qualification work, the research and development of an expert system using a Bayesian inference system was carried out. The work is aimed at solving the problem of uncertainty in expert systems and improving the decision-making process.

The theoretical foundations of Bayesian inference and its application in expert systems are analyzed. The architecture of the expert system is developed, including the knowledge base, the inference system and the user interface. The application of Bayesian inference system is investigated on specific examples and use cases. Software for implementing the expert system and conducting tests is being developed. As a result, an expert system with a Bayesian inference system is obtained, which demonstrates the ability to effectively solve decision-making problems under conditions of uncertainty.

Keywords: expert system, Bayesian system, logical inference, uncertainty, decision-making.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ТА АНАЛІЗ СТАНУ РОЗРОБОК ЕКСПЕРТНИХ СИСТЕМ РОЗВ’ЯЗАННЯ ІНТЕЛЕКТУАЛЬНИХ ЗАДАЧ ..	11
1.1. Поняття, призначення та структура експертної системи.....	11
1.2. Моделі представлення знань та їх порівняльна характеристика	13
1.3. Мови та технології програмування штучного інтелекту	15
РОЗДІЛ 2. ПРОЕКТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ З БАЙЄСІВСЬКОЮ СИСТЕМОЮ ЛОГІЧНОГО ВИСНОВКУ	18
2.1. Постановка задачі на розробку експертної системи охоронної тривожної сигналізації на базі байєсівської мережі довіри.....	18
2.2. Математична модель міркування в байєсівських мережах	20
2.3. Алгоритм розрахунку імовірнісних параметрів експертної системи охоронної тривожної сигналізації	23
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЕКСПЕРТНОЇ СИСТЕМИ З БАЙЄСІВСЬКОЮ СИСТЕМОЮ ЛОГІЧНОГО ВИСНОВКУ	26
3.1. Програмна реалізація експертної системи ОТС	26
3.2. Представлення байєсівської мережі довіри експертної системи ОТС та вхідних її імовірнісних параметрів в пам’яті комп’ютера....	28
3.3. Розробка, налагодження та виконання програмного коду експертної системи ОТС в середовищі пакету SWI-Prolog	32
ВИСНОВКИ	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	54

ВСТУП

Сучасний світ характеризується швидким розвитком інформаційних технологій та зростанням обсягу даних, що потребують аналізу та висновків. У цьому контексті експертні системи здобувають все більшу популярність як інструменти для прийняття рішень та автоматизації експертних знань. Однак, ефективно логічне висновкування в експертних системах є складною задачею, оскільки вона вимагає обробки невизначеності, неповної та суперечливої інформації. У рамках цієї дипломної роботи досліджується експертна система з байєсівською системою логічного висновку, яка є одним із підходів до вирішення проблеми невизначеності в експертних системах. Байєсівський підхід базується на теоремі Байєса та ймовірнісних методах, що дозволяють моделювати ймовірності подій та визначати ймовірності висновків на основі наявної інформації.

Актуальність теми дослідження. Експертні системи є інформаційними технологіями, які моделюють та реплікують експертний досвід та знання в певній галузі. Вони можуть використовуватися для прийняття рішень, вирішення складних проблем або надання порад відповідно до певних правил і логіки. Байєсівські системи логічного висновку ґрунтуються на статистичних методах, зокрема на теорії ймовірностей та теорії байєсового статистичного висновку. Вони дозволяють оцінювати ймовірності різних подій і враховувати нову інформацію для оновлення цих ймовірностей.

Поєднання експертних систем із байєсівською системою логічного висновку відкриває нові можливості для створення більш точних та надійних систем, здатних робити складні висновки на основі доступної інформації. Такі системи можуть бути корисними у багатьох галузях, включаючи медицину, фінанси, інженерію, природні науки та інші.

Мета дипломної роботи: обґрунтування та розробка експертної системи з байєсівською системою логічного висновку (на прикладі системи охоронної тривожної сигналізації).

Об'єктом дослідження є процеси функціонування системи охоронної тривожної сигналізації (ОТС).

Предметом дослідження дипломної роботи є моделі, методи та алгоритми створення експертної системи охоронною тривожної сигналізації з використанням байєсівських мереж.

В ході виконання дипломної роботи були використані такі методи:

- проведено докладне дослідження літератури, пов'язаної з байєсівською системою логічного висновку, експертними системами та суміжними областями. Цей метод дозволив збільшити розуміння теоретичних основ існуючих підходів та методологій;
- проведено аналіз вимог до експертної системи з байєсівською системою логічного висновку. Цей метод допоміг визначити потреби користувачів, функціональні та не функціональні вимоги до системи;
- розроблено архітектуру та дизайн експертної системи з використанням байєсівської системи логічного висновку. Цей метод включав в себе визначення структури бази знань, моделей логічного висновку та інтерфейсу користувача;
- реалізовано експертну систему з використанням певного програмного забезпечення або програмувальних мов. Цей метод включав в себе розробку коду, інтеграцію компонентів системи та налаштування параметрів;
- проведено тестування розробленої системи, щоб перевірити її функціональність, надійність та ефективність. Також ви оцінили результати тестування та порівняли їх з поставленими вимогами;
- аналізовано результати тестування та оцінки системи, визначили її переваги, недоліки та можливості подальшого вдосконалення.

Для досягнення мети необхідно виконати наступні завдання:

- 1) Ознайомитись з принципами та математичними моделями, що лежать в основі байєсівського висновку.
- 2) Розібратись у принципах роботи експертних систем, їх структурі та компонентах.

- 3) Визначити компоненти системи, такі як база знань, система логічного висновку та інтерфейс користувача. Розробити структуру бази знань та встановити зв'язки між різними елементами системи.
- 4) Використати програмне забезпечення або певну програмувальну мову для реалізації байєсівської системи логічного висновку згідно з розробленою архітектурою.
- 5) Виконати ряд тестових сценаріїв для перевірки функціональності та коректності роботи системи. Оцінити ефективність та надійність системи, порівняти результати з поставленими вимогами.
- 6) Проаналізувати отримані результати тестування та оцінки системи. Визначити переваги, недоліки та можливості подальшого вдосконалення системи з байєсівською системою логічного висновку.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається зі вступу, трьох розділів, списку використаних джерел з 56 найменувань та додатків. Містить 50 сторінок основного тексту, рисунків і таблиць

РОЗДІЛ 1.

ТЕОРЕТИЧНІ АСПЕКТИ ТА АНАЛІЗ СТАНУ РОЗРОБОК ЕКСПЕРТНИХ СИСТЕМ РОЗВ'ЯЗАННЯ ІНТЕЛЕКТУАЛЬНИХ ЗАДАЧ

1.1. Поняття, призначення та структура експертної системи

Експертна система є інтелектуальною системою, що використовує знання та інференційні механізми для вирішення складних задач, які зазвичай вимагають експертних знань. Вона спроектована для виявлення причинно-наслідкових залежностей, формулювання висновків, надання рекомендацій та розв'язання проблем, які вимагають професійного аналізу.

Призначення експертної системи полягає в тому, щоб використовувати експертні знання і навички, які зазвичай належать кваліфікованим людям, і перетворювати їх на алгоритмічні процедури та розумові моделі. Експертні системи дозволяють розробникам зберігати, оновлювати та використовувати експертні знання для вирішення різноманітних задач.

Ось таблиця, яка описує основні компоненти структури експертної системи:

Таблиця 1.1

Структура експертної системи - Основні компоненти

Компонент	Опис
База знань	Це центральна складова експертної системи, що містить експертні знання, правила, факти, евристичні правила та інші типи інформації. База знань зберігає експертні знання в структурованій формі, що дозволяє системі логічно висновувати та робити прийняття рішень.
Система управління	Це основний рушій експертної системи, який відповідає за інференцію на основі знань, що містяться в базі знань. Вона застосовує правила та стратегії логічного висновку для отримання нової інформації з вже наявних даних.

Продовження таблиці 1.1

Механізм пояснень	Цей компонент експертної системи відповідає за надання пояснень користувачеві або іншим зацікавленим сторонам щодо висновків, рекомендацій або процесу прийняття рішень, зроблених системою. Він дозволяє роз'яснити логіку та обґрунтованість рішень, прийнятих системою.
Інтерфейс користувача	Це компонент, який забезпечує взаємодію між користувачем та експертною системою. Інтерфейс користувача може бути графічним, текстовим або комбінованим і дозволяє користувачеві вводити дані, отримувати результати, задавати запитання та отримувати пояснення в зручній для нього формі.
База даних	Цей компонент містить вхідні дані, факти, специфікації, моделі, результати аналізу тощо.

База знань є центральною складовою системи, де зберігаються експертні знання, правила, факти та інші типи інформації. Це дозволяє системі логічно висновувати та приймати рішення на основі наявної інформації. Система управління відповідає за інференцію на основі знань, що містяться в базі знань. Вона застосовує правила та стратегії логічного висновку для отримання нової інформації з вже наявних даних. Цей компонент є основним рушієм експертної системи. Механізм пояснень відповідає за надання пояснень користувачеві або іншим зацікавленим сторонам щодо висновків, рекомендацій або процесу прийняття рішень, зроблених системою. Він дозволяє роз'яснити логіку та обґрунтованість рішень, прийнятих системою.

Інтерфейс користувача є компонентом, що забезпечує зручну взаємодію між користувачем та експертною системою. Цей компонент може мати графічний, текстовий або комбінований інтерфейс, дозволяючи користувачеві вводити дані, отримувати результати, задавати запитання та отримувати пояснення в зручній для нього формі.

База даних містить вхідні дані, факти, специфікації, моделі та результати аналізу. Цей компонент є джерелом інформації для експертної системи і допомагає їй виробляти висновки та приймати рішення.

Усі ці компоненти спільно працюють для досягнення мети експертної системи та забезпечення її ефективного функціонування. Вивчення структури експертної системи є важливим етапом розробки та впровадження таких систем для досягнення оптимальних результатів.

Структура типової експертної системи наведена на рис. 1.1.

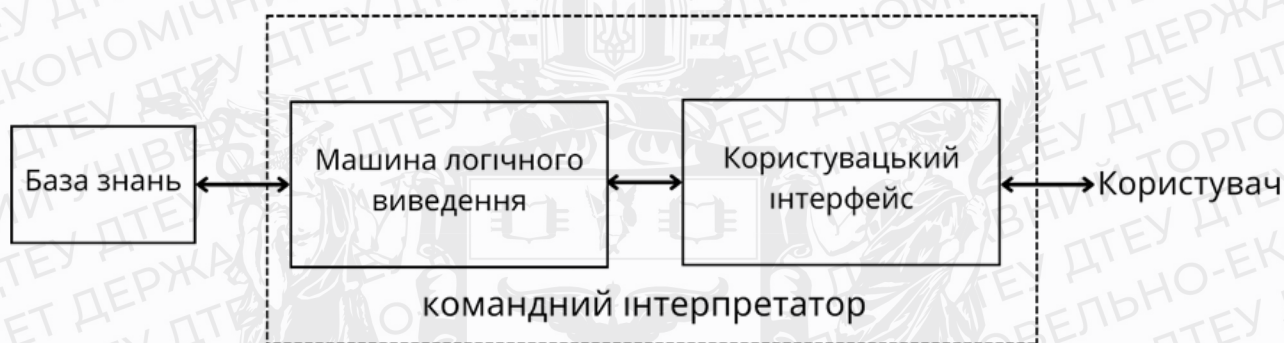


Рис 1.1. Структура типової експертної системи

1.2. Моделі представлення знань та їх порівняльна характеристика

Моделі представлення знань є важливими компонентами експертних систем. Вони визначають способи, якими знання можуть бути представлені та організовані в системі. Кожна модель має свої переваги, недоліки та відповідає певним вимогам та типам знань. Нижче наведено порівняльну характеристику декількох моделей представлення знань:

1. Пропозиційна модель (Propositional Model):

- знання представляються у формі пропозицій або утверджень;
- ця модель досить проста і підходить для висловлення фактів, правил та логічних залежностей;
- обмежена у вираженні складних знань та контекстуальних відношень.

2. Первинна модель фреймів (Primary Frame Model):

- знання представляються у вигляді фреймів, що містять атрибути та значення;
- фрейми дозволяють описати об'єкти, їхні властивості та відношення між ними;
- модель фреймів дозволяє структурувати знання та використовувати ієрархії.

3. Продукційна модель (Production Model):

- знання представляються у вигляді продукційних правил, що містять умови та дії;
- продукційні правила описують логіку висновку та дії, які потрібно виконати;
- ця модель добре підходить для систем, які виконують велику кількість правил.

4. Семантична мережа (Semantic Network):

- знання представляються у вигляді графа, де вузли представляють поняття, а ребра - відношення між ними;
- семантичні мережі дозволяють описати семантичні зв'язки та ієрархії знань;
- використання семантичних мереж може ускладнити процес інференції та обробки знань через необхідність навігації по графу.

5. Логічна модель (Logical Model):

- знання представляються у формі логічних виразів, використовуючи формальну логіку;
- логічна модель дозволяє точно формалізувати знання та проводити логічний розумовий висновок;
- використання логічної моделі може бути складним при роботі з нечіткими або неструктурованими знаннями.

Кожна модель має свої переваги та обмеження і використовується в залежності від конкретних вимог і характеру задачі. У практиці експертних

систем часто використовують комбінацію різних моделей для ефективного представлення та використання знань.

Результати аналізу демонструють, що кожна модель має свої переваги і обмеження, і вибір конкретної моделі залежить від особливостей конкретної задачі та контексту використання експертної системи.

Однією з найпоширеніших моделей є правило-орієнтована модель, яка використовує правила у формі "якщо-то" для опису знань. Ця модель є простою у розумінні і використанні, але може бути обмеженою в розгляді складних взаємозв'язків між знаннями.

Іншою моделлю є семантична мережа, яка використовує графічне представлення знань у вигляді вузлів та зв'язків. Ця модель дозволяє гнучко представляти структуровані знання та взаємозв'язки між ними, але може бути складнішою у використанні та потребує великої кількості зв'язків у великих системах. Також були розглянуті ієрархічна модель, рамкова модель та базова модель знань. Кожна з цих моделей має свої особливості і використовується в певних сферах застосування.

Залежно від потреб і вимог конкретної задачі, можна вибрати найбільш підходящу модель представлення знань для реалізації експертної системи. Важливо враховувати складність знань, гнучкість використання та можливість розширення системи у майбутньому.

1.3. Мови та технології програмування штучного інтелекту

Штучний інтелект (ШІ) використовує різноманітні мови програмування та технології для розробки імплементації алгоритмів та моделей інтелектуального аналізу даних. Деякі з найпопулярніших мов та технологій, що використовуються у сфері штучного інтелекту, включають:

1. **Python:** Python є однією з найпоширеніших мов програмування для штучного інтелекту. Він має велику кількість бібліотек та фреймворків, таких як TensorFlow, PyTorch, scikit-learn, і Keras, які спрощують розробку моделей машинного навчання, нейронних мереж та інших алгоритмів штучного інтелекту.

2. **Java:** Java є широко використовуваною мовою програмування у сфері штучного інтелекту. Вона має ряд фреймворків, таких як Apache Mahout і Weka, які допомагають розробникам використовувати алгоритми машинного навчання та інтелектуального аналізу даних.

3. **R:** R є спеціалізованою мовою програмування для статистичного аналізу та візуалізації даних. Вона має багато пакетів, таких як caret, randomForest і ggplot2, які сприяють розробці моделей машинного навчання та статистичного аналізу.

4. **Lisp:** Lisp є старіючою, але все ще використовуваною мовою програмування, особливо в галузі символічних обчислень та штучного інтелекту. Вона має потужні механізми маніпулювання символічними даними та символічними виразами, що робить її популярною для розробки експертних систем та інших інтелектуальних алгоритмів.

5. **Julia:** Julia є відносно новою мовою програмування, яка поєднує високу продуктивність з простотою використання. Вона набуває популярності в сфері штучного інтелекту завдяки своїм можливостям для числових обчислень, обробки даних та розв'язання задач машинного навчання.

6. **TensorFlow та PyTorch:** TensorFlow та PyTorch є фреймворками глибокого навчання, які надають інструменти для розробки та тренування нейронних мереж. Вони підтримують мови програмування, такі як Python, і забезпечують широкі можливості для роботи зі штучним інтелектом.

Це лише кілька прикладів мов програмування та технологій, які використовуються у сфері штучного інтелекту. Вибір мови залежить від конкретних потреб та вимог проекту, а також від досвіду та вподобань розробника.

Мови та технології програмування для штучного інтелекту є важливим інструментарієм для розробки і розгортання інтелектуальних систем. У даному дослідженні були розглянуті деякі з найпоширеніших мов та технологій, використовуваних у цій області.

- Python відомий своєю простотою, гнучкістю та багатим екосистемою бібліотек для розробки ШІ. Він є популярним вибором для реалізації нейронних мереж, машинного навчання та інших алгоритмів ШІ.
- Java використовується для розробки систем інтелектуального аналізу даних та експертних систем. Вона має широку підтримку інструментів і бібліотек для обробки даних та розробки алгоритмів ШІ.
- C++ відомий своєю продуктивністю та швидкістю, що робить його популярним вибором для реалізації високоефективних алгоритмів штучного інтелекту.
- Prolog – декларативна мова логічного програмування, в основу якої покладений математичний апарат числення предикатів. Найбільш поширена мова штучного інтелекту. Ця мова знайшла розвиток в візуальній об'єктно-орієнтованій та декларативній мові Visual Prolog.

Вибір мови та технологій програмування для ШІ залежить від конкретних потреб та вимог проекту. Необхідно враховувати функціональність, продуктивність, підтримку бібліотек та фреймворків, які допоможуть реалізувати поставлені завдання. Загалом, мови та технології програмування для штучного інтелекту постійно розвиваються і надають розробникам широкий спектр інструментів для створення розумних систем.

Для розробки експертної системи охоронної тривожної сигналізації буде використана мова штучного інтелекту SWI-Prolog.

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ З БАЙЄСІВСЬКОЮ СИСТЕМОЮ ЛОГІЧНОГО ВИСНОВКУ

2.1. Постановка задачі на розробку експертної системи охоронної тривожної сигналізації на базі байєсівської мережі довіри

Мета: Розробити експертну систему охоронної тривожної сигналізації, яка використовує байєсівську мережу довіри для ефективного виявлення потенційних небезпек та спрацьовує тривожний сигнал при необхідності.

Критерії успішності:

1. Ефективне виявлення потенційних небезпек з високою точністю.
2. Мінімізація помилкових сигналів тривоги.
3. Швидка реакція на сигнали тривоги з відповідними заходами безпеки.
4. Легка розширюваність та модифікуваність системи для впровадження нових правил та знань.

Завдання:

1. Зібрати базу знань про потенційні небезпеки, включаючи типи загроз, їх ознаки та характеристики.
2. Розробити модель байєсівської мережі довіри, яка враховує залежності між різними ознаками небезпек та тривожними сигналами.
3. Реалізувати механізми вимірювання та збору даних про стан системи охоронної сигналізації та оточуючого середовища.
4. Розробити алгоритми виявлення потенційних небезпек на основі байєсівської мережі довіри.
5. Реалізувати систему тривожної сигналізації, яка спрацьовує при виявленні небезпек з врахуванням рівня довіри та критичності ситуації.
6. Провести тестування та валідацію системи на різних сценаріях та перевірити її ефективність та надійність.

7. Забезпечити інтерфейс користувача, що дозволяє взаємодіяти з експертною системою, отримувати інформацію про стан системи, конфігурувати параметри та отримувати повідомлення про тривогу.
8. Забезпечити можливість моніторингу та аналізу даних, що збираються системою, для подальшого вдосконалення та оптимізації роботи експертної системи.
9. Здійснити документування системи, включаючи опис бази знань, моделі байєсівської мережі довіри, алгоритми та логіку роботи системи.
10. Після успішного завершення розробки експертної системи провести її впровадження та навчання персоналу з її використання.
11. Забезпечити підтримку та обслуговування системи протягом її експлуатаційного періоду, включаючи виправлення помилок, розширення функціональності та оновлення бази знань.

Система охоронної тривожної сигналізації може бути описана наступними подіями: злом об'єкту, середовище, в якому знаходиться цей об'єкт (гроза), спрацювання датчиків, спрацювання тривоги та посилення сповіщення на телефон.

Структура експертної система охоронної тривожної сигналізації наведена на рис. 2.1.



Рис. 2.1. Експертна система охоронної тривожної сигналізації (ОТС)

Результати: Після успішної розробки та впровадження експертної системи охоронної тривожної сигналізації на базі байєсівської мережі довіри очікується отримання надійного та ефективного засобу для виявлення потенційних небезпек та спрацювання тривожних сигналів у реальному часі. Система повинна забезпечувати високу точність виявлення небезпек та мінімізацію помилкових сигналів тривоги, що дозволить забезпечити безпеку об'єкту, де вона використовується. Крім того, система повинна бути гнучкою, легко модифікуватись та розширюватись для впровадження нових правил та знань.

2.2. Математична модель міркування в байєсівських мережах

Байєсівська мережа є статистичною моделлю, яка використовує теорію ймовірностей для моделювання міркування та висновків. Математична модель міркування в байєсівських мережах базується на теоремі Байєса, яка встановлює зв'язок між умовною ймовірністю та зворотною умовною ймовірністю.

Нехай маємо байєсівську мережу зі змінними X_1, X_2, \dots, X_n , де кожна змінна представляє події або стани, про які ми маємо деяку інформацію. Зв'язки між змінними моделюються за допомогою деревоподібної структури, де кожна змінна є вузлом, а зв'язки між вузлами відображають залежності між подіями.

Математично, міркування в байєсівській мережі полягає у визначенні умовних ймовірностей змінних на основі наявної інформації та залежностей між ними. Для цього використовуються правила обчислення умовних ймовірностей, зокрема, правило множення та правило додавання ймовірностей.

Математична модель міркування в байєсівських мережах може бути представлена у вигляді системи рівнянь, де кожна рівняння відповідає умовній ймовірності змінної на основі умовних ймовірностей її предків. Дані умовні ймовірності можуть бути визначені експертно або зібрані з реальних даних.

Загальні кроки для розрахунку в байєсівській мережі:

1. Визначте входні значення для всіх змінних у мережі.
2. Застосуйте формули байєсівського правила для обчислення умовних ймовірностей. Наприклад, для обчислення умовної ймовірності $P(A|B)$,

використовуйте формулу $P(A|B) = P(A \cap B) / P(B)$, де $P(A \cap B)$ - ймовірність одночасного виконання подій A та B, $P(B)$ - ймовірність події B.

3. Продовжуйте обчислення для всіх змінних у мережі, використовуючи залежності та формули, що визначені в моделі міркування.

4. Перевірте, чи сума ймовірностей всіх можливих значень кожної змінної дорівнює 1. Це важливо, оскільки сума ймовірностей всіх можливих випадків має бути рівною 1.

5. Залежно від конкретної задачі та моделі, можуть бути проведені додаткові розрахунки, такі як прийняття рішень на основі отриманих ймовірностей або виконання додаткових логічних висновків.

Таблиця 2.1

Загальні кроки для розрахунку в байєсівській мережі

Крок	Опис
1.	Визначення структури мережі: Встановлення структури байєсівської мережі, включаючи взаємозв'язки між вузлами та напрямком впливу.
2.	Визначення вхідних значень: Встановлення значень всіх вхідних змінних, які використовуються для розрахунку ймовірностей.
3.	Встановлення параметрів: Визначення ймовірнісних параметрів для кожного вузла мережі, зокрема ймовірностей умовних подій.
4.	Розрахунок умовних ймовірностей: Використання байєсівського правила для обчислення умовних ймовірностей залежно від структури мережі та доступної інформації.
5.	Продовження розрахунків: Продовження обчислення умовних ймовірностей для всіх змінних у мережі, використовуючи залежності та формули, що визначені в моделі міркування.

6.	Перевірка на нормалізацію: Переконавання, що сума ймовірностей всіх можливих значень кожної змінної дорівнює 1.
7.	Використання отриманих ймовірностей: Застосування отриманих ймовірнісних параметрів для логічних висновків, прийняття рішень або прогнозування вірогідних подій.

Ця таблиця надає загальний огляд кроків, які зазвичай виконуються при розрахунку в байєсівській мережі. Зауважте, що кожен конкретний випадок може мати свої унікальні особливості та додаткові кроки в залежності від контексту та вимог проекту.



Рис 2.2. Класифікація моделей за засобами моделювання

Математичну модель байєсівських міркувань можна описати за допомогою наступних пунктів:

1. Теорема Байєса для одної гіпотези та одного свідoctва

$$P(h|e) = \frac{P(h) \cdot P(e|h)}{P(e)} \quad (2.1)$$

де,

$P(h)$ — апіорна (безумовна) ймовірність події (гіпотези) h .

$P(e)$ — апіорна (безумовна) ймовірність події (свідoctва) e .

$P(e | h)$ — апіорна (умовна) ймовірність події (свідoctва) e при умові, що відбулася подія (гіпотеза) h .

$P(h | e)$ — апіорна (умовна) ймовірність події (гіпотези) h при умові, що відбулася подія (свідoctво) e .

2. Формула Байєса, коли є декілька комплексних подій (свідoctв) та декілька подій (гіпотез)

$$p(h_i | e_1 \& e_2 \& \dots \& e_n) = (p(h_i) * p(e_1 \& e_2 \& \dots \& e_n | h_i)) / p(e_1 \& e_2 \& \dots \& e_n) \quad (2.2)$$

3. Особливість формули Байєса полягає в тому, що формули (ймовірності), які знаходяться в її правій частині легше визначити, ніж формулу (ймовірність), яка розташована ліворуч.

4. Складність обчислення формули Байєса.

Для певних задач, наприклад у медицині, кількість гіпотез (захворювань) може бути до 200, а свідoctв (симптомів) — до 2000. В такому випадку прийдеться розглядати більше 800 000 000 ймовірностей. Для спрощення задачі треба будувати байєсівську мережу довіри, при побудові якої припускають, що в більшості випадків між свідoctвами відсутня залежність.

2.3. Алгоритм розрахунку ймовірнісних параметрів експертної системи охоронної тривожної сигналізації

Визначення ймовірності параметрів експертної системи охоронної тривожної сигналізації

Таблиця 2.2.

Алгоритм розрахунку ймовірнісних параметрів експертної системи охоронної тривожної сигналізації

Крок	Опис кроку
1	Визначення вхідних змінних: Встановіть значення всіх вхідних змінних, які впливають на рівень тривоги системи, такі як сенсори, датчики або вхідні дані.
2	Визначення вихідної змінної: Визначте вихідну змінну, яка вказує на

	рівень тривоги системи.
3	Встановлення експертних знань: Зберіть експертні знання про залежності між вхідними та вихідною змінними.
4	Розробка моделі міркування: Розробіть модель міркування, яка описує залежності між вхідними та вихідною змінною.
5	Встановлення ймовірнісних параметрів: Визначте ймовірнісні параметри для кожної залежності у моделі міркування.
6	Розрахунок умовних ймовірностей: Застосуйте байєсівське правило для обчислення умовних ймовірностей.
7	Перевірка на нормалізацію: Переконайтеся, що сума ймовірностей всіх можливих значень кожної змінної дорівнює 1. Це важливо, оскільки сума ймовірностей всіх можливих випадків має бути рівною 1.
8	Використання експертної системи: Використовуйте отриману модель міркування та ймовірнісні параметри для вирішення задачі охоронної тривожної сигналізації.

Ця таблиця містить опис кроків алгоритму розрахунку ймовірнісних параметрів експертної системи охоронної тривожної сигналізації на базі **байєсівської мережі довіри**. Кожен крок пояснює основний етап алгоритму, що допомагає побудувати модель системи та розрахувати ймовірнісні параметри для прийняття рішень.

1. Визначення вхідних змінних: У цьому кроці встановлюються значення всіх вхідних змінних, які впливають на рівень тривоги системи, наприклад, дані з сенсорів або датчиків.
2. Визначення вихідної змінної: Визначається вихідна змінна, яка вказує на рівень тривоги системи.
3. Встановлення експертних знань: У цьому кроці збираються експертні знання про залежності між вхідними та вихідною змінними.
4. Розробка моделі міркування: Розробляється модель міркування, яка описує залежності між вхідними та вихідною змінною. Ця модель може мати форму байєсівської мережі довіри.

5. Встановлення ймовірнісних параметрів: Для кожної залежності у моделі міркування визначаються ймовірнісні параметри, які відображають ступінь впливу вхідних змінних на вихідну змінну.

6. Розрахунок умовних ймовірностей: Застосовуються байєсівські правила для обчислення умовних ймовірностей, використовуючи встановлені ймовірнісні параметри та експертні знання.

7. Перевірка на нормалізацію: Перевіряється, чи сума ймовірностей всіх можливих значень вихідної змінної дорівнює 1. Це важливо, оскільки сума ймовірностей всіх можливих випадків має бути рівною 1.

8. Використання отриманих ймовірностей: Отримані ймовірнісні параметри можуть бути використані для прийняття рішень щодо рівня тривоги системи або для прогнозування вірогідних подій.

Ця таблиця надає загальний огляд алгоритму розрахунку імовірнісних параметрів експертної системи охоронної тривожної сигналізації на базі байєсівської мережі довіри. Кожен крок важливий для успішного розробки та ефективного використання такої системи.

РОЗДІЛ 3.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЕКСПЕРТНОЇ СИСТЕМИ З БАЙЄСІВСЬКОЮ СИСТЕМОЮ ЛОГІЧНОГО ВИСНОВКУ

3.1. Програмна реалізація експертної системи ОТС

Програмна реалізація експертної системи ОТС (охоронна тривожна сигналізація) може бути виконана з використанням різних технологій та мов програмування. Основними компонентами програмної реалізації є:

1. База знань: Створення та управління базою знань, яка містить експертні правила, факти та інформацію про систему ОТС. База знань може бути реалізована у вигляді бази даних або спеціалізованої системи керування знаннями.
2. Інтерфейс користувача: Розробка інтерфейсу, який дозволяє користувачам взаємодіяти з експертною системою. Інтерфейс може бути реалізований у вигляді веб-додатку, десктопного додатку або мобільного додатку.
3. Інтерпретація правил: Розробка механізму для інтерпретації експертних правил, які описують логіку прийняття рішень в системі ОТС. Цей механізм виконує логічні операції та обробку фактів, щоб визначити рівень тривоги та вжити відповідні заходи.
4. Обробка вхідних даних: Забезпечення можливості отримання вхідних даних від сенсорів, датчиків або інших джерел. Ці дані можуть бути використані для аналізу та визначення рівня тривоги.
5. Виведення результатів: Відображення результатів прийняття рішень користувачу. Це може бути у вигляді відповідного повідомлення, активації сигналізаційних пристроїв або надсилання сповіщень.
6. Модуль тестування: Розробка модуля для тестування експертної системи, який дозволяє перевірити правильність роботи системи та виявити помилки чи недоліки.

7. Інтеграція зі зовнішніми системами: Забезпечення можливості інтеграції з іншими системами, наприклад, системами відеоспостереження, контролем доступу або системами керування будівлею.

Програмна реалізація експертної системи ОТС може бути розроблена з використанням мов програмування, таких як **Python** (Додаток А), **Java** (Додаток Б), **C++**, або з використанням спеціалізованих інструментів та фреймворків, які спрощують розробку експертних систем. При розробці важливо враховувати специфіку системи ОТС, вимоги до продуктивності, масштабованості та надійності.

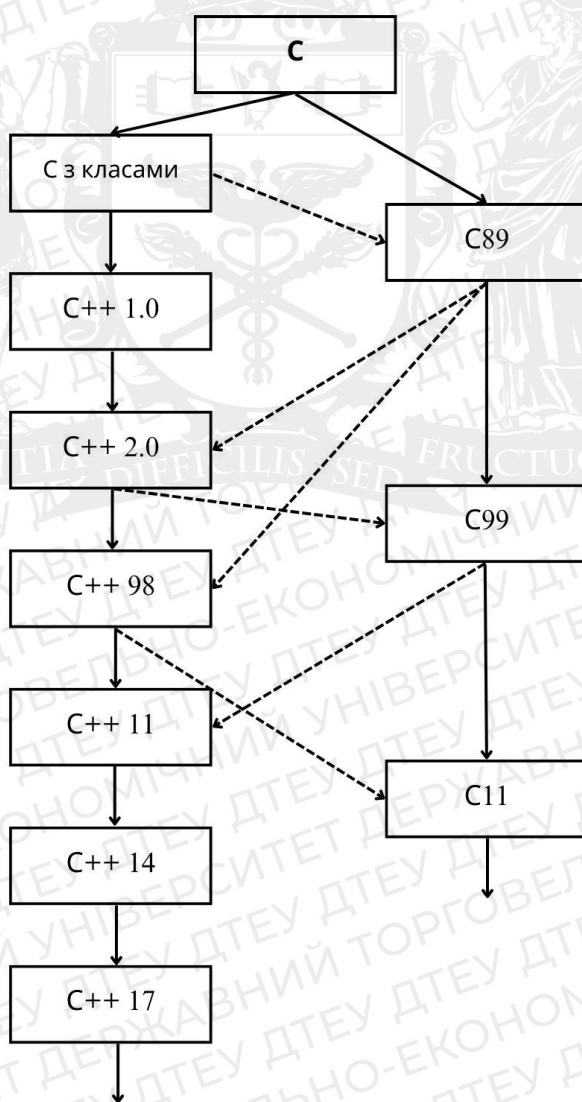


Рис 3.1. Використані мови програмування C++ для розробки програмної реалізації експертної системи ОТС

Представлення байєсівської мережі довіри експертної системи ОТС та вхідних її імовірнісних параметрів в пам'яті комп'ютера

Представлення байєсівської мережі довіри та її імовірнісних параметрів в пам'яті комп'ютера включає в себе використання структур даних для збереження та організації інформації. Основні елементи для представлення байєсівської мережі та її імовірнісних параметрів включають:

Графічне представлення мережі: байєсівська мережа може бути представлена у вигляді графу, де вузли відповідають змінним, а ребра показують залежності між змінними. У пам'яті комп'ютера граф може бути представлений за допомогою структури даних, такої як наприклад списки суміжності або матриці суміжності.

1. Збереження ймовірнісних параметрів: Для кожної змінної у байєсівській мережі потрібно зберігати ймовірнісні параметри, такі як ймовірності умовних та безумовних подій. Ці параметри можуть бути збережені у вигляді таблиць, матриць або інших структур даних, що дозволяють ефективно доступатись до них та виконувати операції обчислення.

2. Механізм доступу до даних: Для ефективного доступу до імовірнісних параметрів та виконання розрахунків, необхідно мати механізм доступу до даних у пам'яті комп'ютера. Це може бути реалізовано через структури даних, функції або класи, які забезпечують зручний інтерфейс для роботи з імовірнісними параметрами.

3. Серіалізація та збереження: Для збереження байєсівської мережі та її імовірнісних параметрів між сеансами роботи може бути використана серіалізація, яка дозволяє перетворити структури даних на послідовність байтів, що може бути збережена на диск або передана по мережі. При необхідності, збережені дані можуть бути відновлені та використані пізніше.

Це лише загальні принципи представлення байєсівської мережі довіри та її імовірнісних параметрів в пам'яті комп'ютера. Конкретні деталі і реалізація будуть залежати від мови програмування, використовуваної платформи та потреб системи (Додаток В).

Таблиця 3.1

Представлення байєсівської мережі довіри та її параметрів в пам'яті

Елемент	Опис
Граф мережі	Графічне представлення мережі з вузлами, що відповідають змінним, та ребрами, що показують залежності між змінними.
Ймовірнісні таблиці	Таблиці, що містять ймовірності умовних та безумовних подій для кожної змінної у мережі. Кожен рядок таблиці представляє один стан змінної, а стовпці відповідають значенням сусідніх змінних.
Структури даних	Використання структур даних, таких як списки суміжності або матриці, для збереження графа мережі та ймовірнісних таблиць.
Алгоритми обчислення	Використання алгоритмів для доступу до даних у мережі, обчислення умовних ймовірностей та виконання інших операцій.
Серіалізація та збереження	Механізми для збереження представлення мережі та її параметрів на диск або передачі по мережі. Це може включати серіалізацію структур даних та використання файлових форматів для збереження.
Завантаження та відновлення	Можливість завантажувати збережені дані мережі та її параметрів збережених на диску та відновлювати роботу експертної системи.

Ця таблиця надає загальний огляд аспектів представлення байєсівської мережі довіри та її ймовірнісних параметрів в пам'яті комп'ютера. Конкретні деталі та реалізація можуть варіюватися залежно від використовуваної мови програмування та інструментарію.

Структура даних для представлення байєсівської мережі довіри та її ймовірнісних параметрів в пам'яті комп'ютера наведена в таблиці 3.2.

Таблиця 3.2

Таблиця представлення байєсівської мережі довіри та ймовірнісних параметрів в пам'яті комп'ютера

Змінна	Стани	Ймовірнісні параметри
X1	x1, x2, x3, ...	$P(X1=x1)$, $P(X1=x2)$, $P(X1=x3)$, ...
X2	y1, y2, y3, ...	$P(X2=y1)$, $P(X2=y2)$, $P(X2=y3)$, ...
X3	z1, z2, z3, ...	$P(X3=z1)$, $P(X3=z2)$, $P(X3=z3)$, ...
...

У цій таблиці кожен рядок відповідає окремій змінній у байєсівській мережі довіри. Стовпці "Стани" містять можливі стани або значення цієї змінної, наприклад, x_1 , x_2 , x_3 і так далі. Стовпці "Імовірнісні параметри" містять відповідні ймовірності для кожного стану змінної.

Наприклад, якщо ми маємо змінну X_1 з трьома можливими станами x_1 , x_2 , x_3 , то для кожного стану ми маємо відповідні ймовірності $P(X_1=x_1)$, $P(X_1=x_2)$, $P(X_1=x_3)$. Аналогічно, для інших змінних X_2 , X_3 і так далі, ми маємо відповідні ймовірності для кожного стану.

Ця таблиця може бути реалізована у вигляді масиву або структури даних у пам'яті комп'ютера, де кожен рядок таблиці представляє окремий об'єкт, яка містить інформацію про змінну, її стани та відповідні ймовірнісні параметри.

Це лише один з можливих способів представлення байєсівської мережі довіри та ймовірнісних параметрів в пам'яті комп'ютера, інші структури даних або формати також можуть використовуватися залежно від вимог програмної реалізації та використання експертної системи.

Виходячи з вище сказаного, програмне забезпечення експертної системи ОТС було розроблено в інтегрованому середовищі SWI-Prolog (див. рис. 3.2) та воно складається з двох частин: специфікації байєсівської мережі та інтерпретатору байєсівської мережі (див. рис. 2.1).

```
SWI-Prolog (Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 32 bits, version 8.4.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).
?-
Warning: g:/paswiprolog/interpretator_bayes_networks.pl:32:
Warning: Singleton variables: [Cond]
g:/paswiprolog/interpretator_bayes_networks.pl compiled 0.02 sec, 31 clauses
?-
|
```

Рис. 3.2. Інтегроване середовище SWI-Prolog.

3.2. Представлення байєсівської мережі довіри експертної системи ОТС та вхідних її імовірнісних параметрів в пам'яті комп'ютера

Байєсівська мережа довіри охоронної тривожної сигналізації представлена на рис. 3.3.

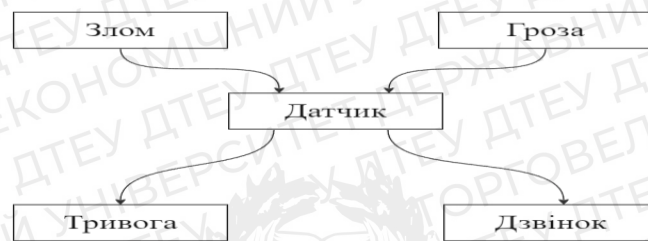


Рис. 3.3. Байєсівська мережа довіри охоронної тривожної сигналізації

Після злому і проникнення в будинок сторонньої особи, ймовірно, відбувається активізація датчика. Припускається, що датчик активізує звуковий тривожний сигнал та автоматичний телефонний дзвінок з попереджувальним повідомленням. Датчик може також спрацювати під дією сильної грози.

Для представлення байєсівської мережі довіри ОТС в пам'яті комп'ютера задіяний предикат `parent`, який має два аргументи, кожен з яких визначає відповідно початковий та кінцевий вузли нашої мережі. Для опису мережі було використано чотири таких предиката (див. рис. 3.4).

```
parent(burglary, sensor).  
parent(lightning, sensor).  
parent(sensor, alarm).  
parent(sensor, call).
```

Рис. 3.4. Предикати представлення байєсівської мережі довіри охоронної тривожної сигналізації в пам'яті комп'ютера. Наведений фрагмент програми визначається у складі бази знань в специфікації мережі.

Крім того, в специфікації мережі бази знань визначаються початкові ймовірності значення, які характеризують ймовірності подій визначених у вузлах мережі (див. рис. 3.5).

$p(\text{burglary}, 0.001)$.
 $p(\text{lightning}, 0.02)$.
 $p(\text{sensor}, [\text{burglary}, \text{lightning}], 0.9)$.
 $p(\text{sensor}, [\text{burglary}, \text{not lightning}], 0.9)$.
 $p(\text{sensor}, [\text{not burglary}, \text{lightning}], 0.1)$.
 $p(\text{sensor}, [\text{not burglary}, \text{not lightning}], 0.001)$.
 $p(\text{alarm}, [\text{sensor}], 0.95)$.
 $p(\text{alarm}, [\text{not sensor}], 0.001)$.
 $p(\text{call}, [\text{sensor}], 0.9)$.
 $p(\text{call}, [\text{not sensor}], 0.0)$.

Рис. 3.5. Початкові ймовірнісні показники байесівської мережі ОТС.

3.3. Розробка, налагодження та виконання програмного коду експертної системи ОТС в середовищі пакету SWI-Prolog

Розробка, налагодження та виконання програмного коду експертної системи ОТС в середовищі пакету SWI-Prolog включає наступні кроки:

1. Встановлення середовища SWI-Prolog: Завантажте та встановіть пакет SWI-Prolog з офіційного веб-сайту SWI-Prolog.
2. Визначення бази знань: Створіть файл з програмним кодом, що містить базу знань для експертної системи ОТС. В цьому файлі ви можете описати правила, факти та логічні залежності, які використовуються для прийняття рішень у системі.
3. Розробка правил: Визначте правила, які використовуються для міркування та прийняття рішень у системі. Використовуйте логічні вирази та умови, щоб описати логіку системи та її поведінку.
4. Виконання програмного коду: Запустіть SWI-Prolog та завантажте розроблений файл з базою знань та правил. Виконайте програмний код, щоб активувати експертну систему та дозволити їй приймати запити та здійснювати логічні висновки.

5. Тестування та налагодження: Перевірте працездатність експертної системи шляхом введення тестових запитів та перевірки результатів. Виконайте необхідні налагодження, якщо є помилки або неправильні результати.

6. Оптимізація та розширення: При потребі оптимізуйте програмний код для поліпшення продуктивності та швидкодії системи. Розширюйте систему шляхом додавання нових правил, фактів або зміни існуючих логічних залежностей.

7. Документація: Забезпечте належну документацію для програмного коду експертної системи. Описуйте правила, базу знань, вхідні та вихідні параметри, щоб інші користувачі могли розуміти та використовувати систему.

8. Використання та підтримка: Розгорніть експертну систему ОТС на потрібному середовищі та забезпечуйте її підтримку та обслуговування. Переконайтеся, що система працює згідно з очікуваннями та надає необхідні результати.

Використання середовища пакету SWI-Prolog дозволяє реалізувати експертну систему ОТС за допомогою мови програмування Prolog, яка має вбудовану підтримку логічного програмування та міркування. Це дозволяє ефективно моделювати логічні залежності та приймати рішення на основі правил та фактів, що представлені у базі знань (Додаток Г).

Компоненти експертної системи включають наступні елементи:

1. База знань: Це центральний компонент експертної системи, який містить знання експерта з конкретної предметної області. База знань може бути представлена у вигляді правил, фактів, процедур або графів залежностей. Ці знання використовуються системою для прийняття рішень та висновків.

2. Інтерфейс користувача: Цей компонент дозволяє взаємодіяти з експертною системою. Він може бути графічним, текстовим або голосовим і дозволяє користувачеві задавати питання, вводити дані, отримувати рекомендації або пояснення.

3. Механізм інтерпретації: Цей компонент відповідає за обробку запитів користувача та виконання логічного висновку на основі знань, що містяться в базі

знань. Він може використовувати різні алгоритми логічного висновку, включаючи байєсівський підхід.

4. Механізм пояснень: Цей компонент надає користувачу пояснення результатів і рекомендацій, що надаються експертною системою. Він може розкривати логіку висновку, вказувати на причини прийняття рішення або надавати додаткову інформацію про обґрунтування рекомендацій.

5. Система управління знаннями: Цей компонент відповідає за зберігання, оновлення та управління знаннями, що містяться в базі знань. Він може включати інструменти для редагування правил, внесення нових фактів, моніторингу та аналізу використання знань.

6. Механізм перевірки достовірності: Цей компонент відповідає за перевірку достовірності та валідності введених даних. Він може використовувати правила або алгоритми перевірки, щоб переконатися, що введені дані відповідають вимогам системи та предметної області.

7. Механізм вирішення конфліктів: Цей компонент відповідає за вирішення ситуацій конфлікту, коли правила або факти в базі знань суперечать одне одному або мають різні ступені впевненості. Він може використовувати стратегії пріоритетизації, вагові коефіцієнти або інші методи для прийняття рішення в таких ситуаціях. Механізм моніторингу та оновлення: Цей компонент відповідає за постійний моніторинг результатів експертної системи та оновлення бази знань. Він може включати механізми для збору даних про ефективність системи, виявлення нових знань, оцінки точності висновків і покращення якості роботи системи.

8. Модуль навчання: В деяких експертних системах може бути присутній окремий компонент для навчання системи. Цей модуль може використовувати методи машинного навчання для автоматичного вивчення знань на основі вхідних даних і результатів системи. Навчання може включати адаптацію правил, ваг та параметрів моделей для покращення результатів.

9. Модуль визначення неясності: У деяких випадках експертна система може використовувати модуль визначення неясності для розуміння та врахування

неявних знань або вимог користувача. Цей компонент може аналізувати контекст, інтерпретувати неявні дії або побажання, а також використовувати відповідні методи для врахування цих аспектів у процесі логічного висновку.

10. Модуль планування: В деяких випадках експертна система може включати модуль планування для визначення послідовності дій або рекомендавання оптимального рішення. Цей модуль може аналізувати доступні ресурси, враховувати обмеження і пріоритети, а також застосовувати стратегії планування для досягнення поставленої мети.

Взаємодія цих компонентів дозволяє експертній системі з байєсівською системою логічного висновку здійснювати аналіз даних, виконувати логічне висновку, надавати рекомендації та пояснення, а також керувати та оновлювати базу знань для постійного вдосконалення системи.

Задача побудови бази знань є однією з найскладніших у розробці експертної системи. Ця база знань включає розробку та заповнення структури, і процес її формування є інтерактивним, з можливими поверненнями до попередніх етапів. Одним із найскладніших аспектів цього процесу є здобуття знань від експертів та їх подальша формалізація.

У системі в будь-який момент часу існують три типи знань. Перші - структуровані знання, які є статичними і не змінюються. Другі - структуровані динамічні знання, які змінюються залежно від нової інформації. Треті - робочі знання, які використовуються для розв'язання конкретних задач або надання консультацій. В інформаційній моделі експертної системи основними формами подання знань є логіка предикатів, семантичні мережі та фрейми.

Логіка предикатів використовує мову математичної логіки для формального опису понять та зв'язків між ними у вигляді фактів та правил. Факти є доведеними твердженнями про об'єкти експертної системи, а правила мають форму "ЯКЩО <умова>, ТО <висновок>". У семантичних мережах об'єкти системи представлені вершинами, а відносини між ними - дугами. Фрейми використовуються для опису стереотипних ситуацій і складаються з характеристик та їх значень.

Кожен з цих підходів до подання знань має свої переваги та особливості. Логіка предикатів дозволяє формалізувати знання у математичній формі, логічній формі, семантичні мережі забезпечують наочне відображення відносин між об'єктами, а фрейми дозволяють моделювати стереотипні ситуації з їх характеристиками та значеннями.

Кожен з цих підходів має свої переваги та особливості. Логіка предикатів дозволяє формалізувати знання у математичній формі, що спрощує конструкцію експертної системи, дозволяє модульно будувати систему та змінювати та розширювати базу знань. Семантичні мережі забезпечують наочне відображення знань, що є корисним для розуміння складних відносин у системі. Фрейми дозволяють явно моделювати родовиту ієрархію понять та описувати стереотипні ситуації з їх характеристиками.

Крім того, ці підходи можуть використовуватися разом для побудови експертних систем з байєсівською системою логічного висновку. Байєсівська система логічного висновку використовує ймовірнісні методи для оцінки ймовірності вірності висновків на основі доступних даних та знань.

Наприклад, можна використовувати логіку предикатів для формалізації бази знань, семантичні мережі для наочного представлення відносин між об'єктами та фрейми для опису стереотипних ситуацій. Байєсівська система логічного висновку може використовувати ймовірності, отримані з експертних знань та статистичних даних, для здійснення висновків та прийняття рішень.

Таке поєднання різних підходів може забезпечити більш гнучку та потужну експертну систему, яка здатна досягати точних та надійних висновків на основі доступних знань та даних.

Механізм байєсівського логічного висновку.

Механізм байєсівського логічного висновку є центральним елементом експертних систем, що базуються на байєсівському підході. Він використовує статистичні методи та теорію ймовірностей для визначення ймовірності наявності певного стану чи події, заснованої на наявних даних та знаннях.

Основні кроки механізму байєсівського логічного висновку включають:

1. Визначення вхідних даних: Механізм отримує вхідні дані, які можуть бути у вигляді спостережень, фактів або запитань. Наприклад, можуть бути задані певні спостереження про симптоми хвороби.
2. Побудова моделі: Наступним кроком є побудова байєсівської мережі або ймовірнісного графа, який відображає залежності між різними змінними або подіями. Модель включає в себе вузли, які представляють змінні, та дуги, які вказують на ймовірнісні зв'язки між ними.
3. Визначення апіорних ймовірностей: Для кожного вузла в моделі потрібно встановити апіорні ймовірності, які відображають початкові знання про ймовірності подій без урахування нових даних. Ці ймовірності можуть бути встановлені на основі експертних думок, історичних даних або інших джерел.
4. Оновлення ймовірностей з урахуванням нових даних: Після отримання нових даних механізм оновлює ймовірності у моделі, використовуючи теорему байєсівського правила. Це дозволяє перерахувати ймовірності наявності певних станів чи подій з урахуванням нової інформації.
5. Логічний висновок: За допомогою оновлених ймовірностей механізм може здійснити логічний висновок щодо стану чи події, яка виводиться з наявних даних. Наприклад, система може зробити висновок про діагноз хвороби на основі спостережень про симптоми.

Цей механізм дозволяє експертній системі оцінювати ймовірності різних станів та подій на основі доступних даних і знань, а також робити логічні висновки. Він допомагає системі приймати рішення в умовах невизначеності та неповної інформації (див. рис. 3.6).

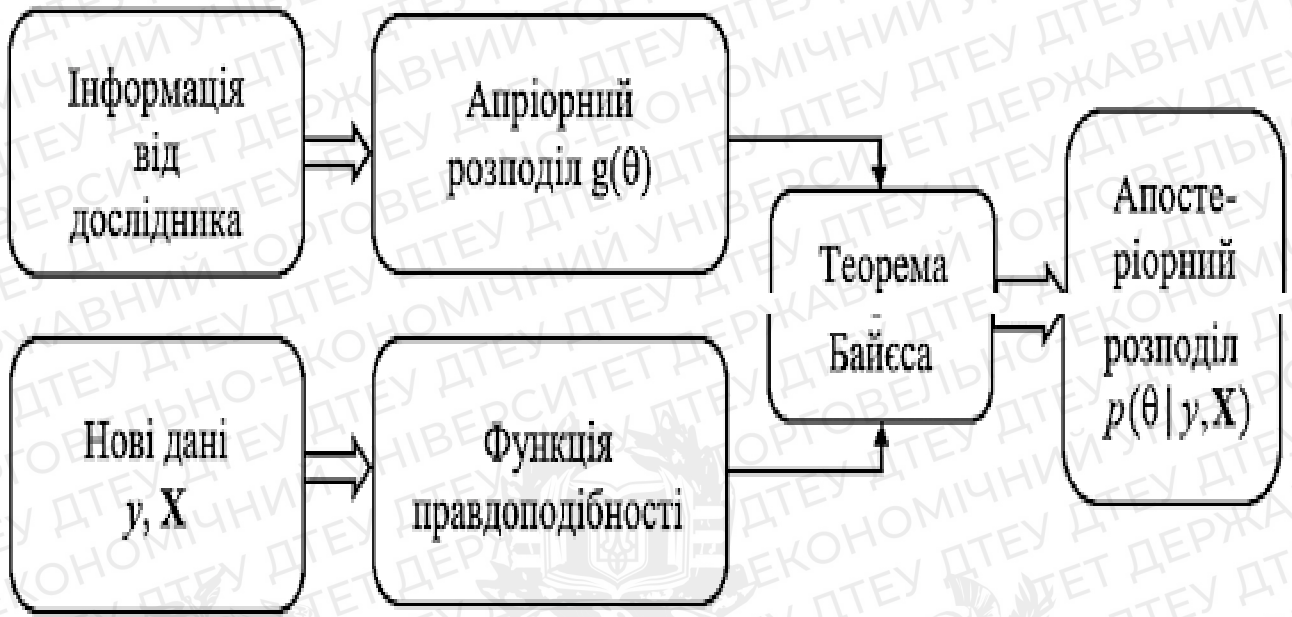


Рис 3.6. Механізм для оцінки ймовірностей станів та подій

Примітка: При роботі з байєсівським логічним висновком слід враховувати, що точність результатів може залежати від якості вхідних даних, точності апріорних ймовірностей та самої моделі. Також можуть бути важливими питання щодо розрахунку і оновлення ймовірностей великих байєсівських мереж, що можуть потребувати ефективних алгоритмів та обчислювальних ресурсів.

Розглянемо приклад діагностичної експертної системи для виявлення несправностей в автомобілі. У нашому прикладі ми маємо чотири можливі несправності, які можуть відбутися в автомобілі: несправна акумуляторна батарея, відсутність палива, проблеми з запалюванням та замаслені свічки. Також ми маємо набір симптомів, які можуть вказувати на ці несправності, такі як негорящі фари, показник палива на нулі, незапускання двигуна, проблеми зі стартером та нестабільна робота двигуна. У нашому прикладі також наведені апріорні ймовірності та умовні ймовірності, які відображають ймовірність виявлення симптомів при наявності певних несправностей.

Таблиця 3.3

Байєсівський аналіз даних

Номер	Подія	Ймовірність (P)	Попередня інформація (A)	Поступова інформація (B)	Факт (AB)	Оновлена ймовірність (P')
1	Подія A1	P(A1)	-	-	-	
2	Подія A2	P(A2)	-	-	-	
3	Подія B1	-	-	P(B1)	-	
4	Подія B2	-	-	P(B2)	-	P(B2)
5	Подія B3	-	-	P(B3)	-	P(B3)
6	Подія AB1	P(AB1)	P(A1)	P(B1)	P(A1B1)	P(AB1)
7	Подія AB2	P(AB2)	P(A2)	P(B2)	P(A2B2)	P(AB2)
8	Подія AB3	P(AB3)	P(A2)	P(B3)	P(A3B3)	P(AB3)
9	Подія AB4	P(AB4)	P(A1)	P(B3)	P(A1B3)	P(AB4)
10	Подія AB5	P(AB5)	P(A1)	P(B1)	P(A1B1)	P(AB5)

У таблиці наведені різні події, ймовірності їх відбуття (P), попередня та поступова інформація (A та B), факти (AB) і оновлені ймовірності (P'). Кожна подія має свій номер для зручності ідентифікації. Події A1 та A2 відповідають попередній інформації, а події B1, B2 та B3 - поступовій інформації. Події AB1, AB2, AB3, AB4 та AB5 представляють факти, які отримані з комбінації попередньої та поступової інформації. Оновлені ймовірності P' відображають актуалізовані значення ймовірностей після отримання фактів.

Таблиця використовується для організації та обчислення ймовірностей в рамках байєсівського логічного висновку. Вона допомагає визначити оновлені ймовірності подій на основі початкових ймовірностей та накопиченої інформації.

У таблиці маємо різні стовпці та рядки, що відображають різні аспекти аналізу. Основні елементи таблиці включають:

1. **Номер:** Це унікальний ідентифікатор кожної події, який допомагає в її ідентифікації та посиланні на неї в інших частинах аналізу.
2. **Подія:** Цей стовпець містить опис кожної події, що аналізується. В даному випадку, це можуть бути події, пов'язані з несправностями автомобіля та симптомами цих несправностей.
3. **Ймовірність (P):** Цей стовпець відображає початкову ймовірність кожної події, яка може бути визначена на підставі попередніх знань або статистичних даних.
4. **Попередня Інформація (A):** Цей стовпець вказує попередню інформацію, яка може бути використана для оцінки ймовірності подій. В даному випадку, це можуть бути значення апріорних ймовірностей.
5. **Поступова Інформація (B):** Цей стовпець представляє поступову інформацію, яка надходить під час аналізу. Це можуть бути спостереження, вимірювання або інші факти, що впливають на оцінку ймовірностей.
6. **Факт (AB):** Цей стовпець відображає комбінацію попередньої та поступової інформації, що використовується для оновлення ймовірностей.
7. **Оновлена Ймовірність (P')**: Цей стовпець містить актуалізовані ймовірності після отримання фактів. Вони розраховуються на основі байєсівського правила висновку.

Ця таблиця дозволяє систематизувати і виконати обчислення з байєсівським логічним висновком.

Запуск програмного забезпечення експертної системи ОТС в середовищі виконується за допомогою введення запиту на розрахунок вірогідності з ім'ям `prob` та визначенням в ньому трьох параметрів: перші з них визначають назви подій, а останній змінну, в яку записується розрахована вірогідність (див. ліве вікно рис.3.7).

Фрагмент програмного забезпечення відображений у вікні редактору SWI-Prolog (див. праве вікно рис.3.7).

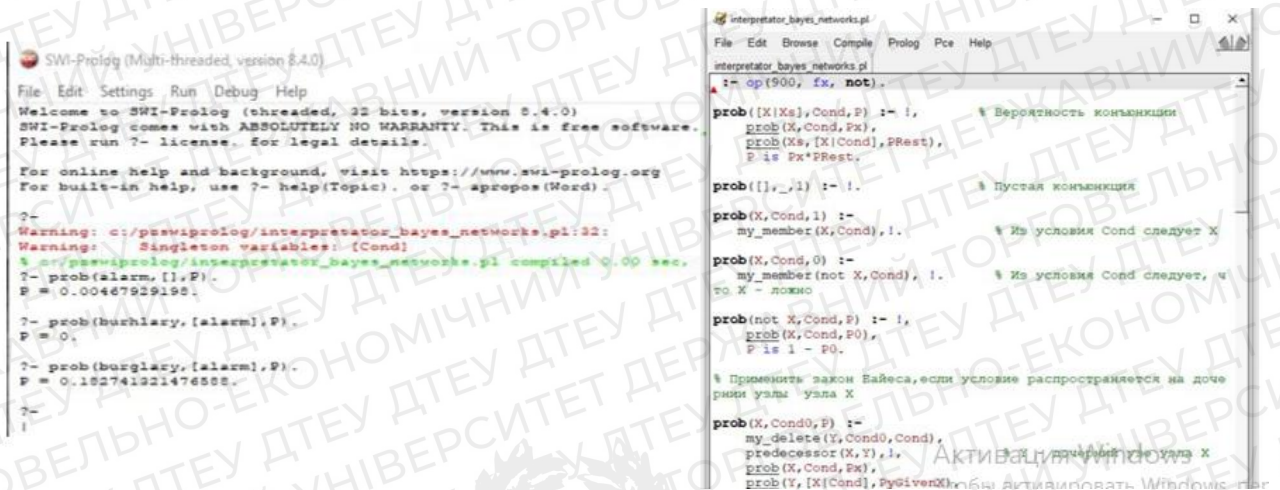


Рис. 3.7. Інтегроване середовище SWI-Prolog з запитами на визначення вірогідностей та результати. Вікно редактору SWI-Prolog з фрагментом програми експертної системи ОТС.

Тестування та налагодження програмного забезпечення експертної системи ОТС.

Тестування та налагодження є важливими етапами у процесі розробки програмного забезпечення. Ці етапи допомагають виявити та усунути дефекти, перевірити відповідність вимогам та забезпечити належну функціональність системи. Тестування включає проведення різних видів тестів з метою перевірки різних аспектів програмного забезпечення, таких як функціональність, продуктивність, надійність та безпека. Деякі з основних видів тестування включають:

1. Функціональне тестування: Перевірка функціональності програмного забезпечення, щоб забезпечити, що воно працює згідно з вимогами та очікуваними результатами.
2. Інтеграційне тестування: Валідація взаємодії між різними компонентами або модулями системи, щоб переконатися, що вони працюють разом належним чином.
3. Регресійне тестування: Перевірка функціональності системи після внесення змін або виправлень для забезпечення того, що ніякі нові дефекти не виникли.

4. Видавниче тестування: Валідація системи в середовищі, яке максимально наближене до реального використання, щоб переконатися, що вона працює належним чином у реальних умовах.

5. Виділене тестування: Перевірка системи на максимальне навантаження для визначення її продуктивності та шкалювання.

Після виявлення дефектів під час тестування, налагодження стає наступним кроком. Це процес виправлення дефектів та оптимізації системи. Розробники аналізують звіти про дефекти та приступають до їх виправлення. Налагодження також може включати оптимізацію швидкодії, виправлення помилок у програмному коді та впровадження додаткових функцій або змін.

Після завершення тестування та налагодження системи, вона готова до випуску та використання. Процес тестування та налагодження допомагає забезпечити якість програмного забезпечення та задовільнити користувачів.

Налагодження проекту в розробці програмного забезпечення є важливою складовою успішного виконання проекту. Воно включає планування, організацію, контроль і керування різними аспектами проекту для досягнення мети в зазначені терміни та в межах бюджету. Деякі ключові елементи налагодження проектом в розробці програмного забезпечення включають:

1. Планування проекту: Визначення мети проекту, обсягу робіт, часового графіка, розподілу ресурсів та інших ключових параметрів. Планування допомагає визначити стратегію виконання проекту та встановити базову лінію для контролю.

2. Організація команди: Формування та управління командою розробників, тестувальників, аналітиків та інших фахівців, які беруть участь у проекті. Це включає розподіл завдань, встановлення комунікаційних ланцюгів та забезпечення ефективної співпраці між учасниками.

3. Виконання проекту: Виконання робіт відповідно до плану, забезпечення виконання вимог, встановлення механізмів контролю прогресу, виявлення та вирішення проблем, які виникають під час виконання проекту.

4. Керування ризиками: Ідентифікація потенційних ризиків, оцінка їх впливу на проект, розробка стратегій мінімізації ризиків та планування дій у разі їх виникнення. Керування ризиками допомагає зменшити негативні наслідки та забезпечити успішне завершення проекту.

5. Контроль і звітність: Встановлення механізмів контролю прогресу проекту, перевірка виконання завдань у встановлені терміни та відповідно до якості, складання звітів про прогрес та представлення їх стейк-холдерам.

Налагодження проекту в розробці програмного забезпечення допомагає забезпечити організацію, ефективне використання ресурсів та досягнення мети проекту.

Таблиця 3.5

Огляд видів тестування

Вид тестування	Опис
Функціональне	Перевірка функціональності окремих модулів або системи
Навантажувальне	Тестування продуктивності та стабільності системи
Санітарне	Тестування для виявлення потенційних проблем безпеки
Регресійне	Перевірка, чи не вплинули зміни на вже налагоджений код
Юніт-тестування	Тестування окремих функцій або компонентів програми
Інтеграційне	Перевірка взаємодії між компонентами системи
Системне	Тестування системи в цілому
Приймальне	Перевірка, чи відповідає система вимогам замовника
Смоук-тестування	Базове тестування для перевірки основних функцій системи
Апаратне	Тестування апаратних компонентів системи
Відмовостійке	Тестування, спрямоване на виявлення помилок та відмов системи

Опис видів тестування, які представлені в таблиці:

1. Функціональне тестування: Це вид тестування, яке спрямоване на перевірку функціональності окремих модулів або системи в цілому. Воно включає перевірку правильності реалізації функцій, обробку вхідних даних та видачу очікуваних результатів.
2. Навантажувальне тестування: Це вид тестування, яке оцінює продуктивність та стабільність системи при підвищеному навантаженні. Його метою є виявлення можливих проблем з продуктивністю, перевантаженням ресурсів та виявленням буття системи.
3. Санітарне тестування: Це вид тестування, яке спрямоване на виявлення потенційних проблем безпеки системи. Воно включає перевірку вразливостей, недоліків у забезпеченні безпеки та можливих шляхів атак.
4. Регресійне тестування: Це вид тестування, яке перевіряє, чи не вплинули зміни в системі на вже налагоджений код. Воно дозволяє виявити непередбачені наслідки внесених змін та переконатися, що функціональність, яка раніше працювала правильно, не постраждала.
5. Юніт-тестування: Це вид тестування, яке спрямоване на перевірку окремих функцій або компонентів програми. Воно виконується на ранніх етапах розробки для перевірки правильності роботи окремих частин системи.
6. Інтеграційне тестування: Це вид тестування, яке перевіряє взаємодію між компонентами системи. Воно включає тестування інтерфейсів, передачу даних та правильну взаємодію між різними модулями.
7. Системне тестування: Це вид тестування, яке оцінює систему в цілому. Воно включає тестування всіх функцій та компонентів, перевірку взаємодії між ними та перевірку відповідності вимогам замовника.
8. Приймальне тестування: Це вид тестування, яке перевіряє, чи відповідає система вимогам замовника. Воно проводиться з метою підтвердження,

що система відповідає вимогам, і включає тестування функцій, продуктивності та інших критеріїв.

9. Смоук-тестування: Це базове тестування, яке перевіряє основні функції системи або компонентів, щоб переконатися, що вони працюють правильно. Воно зазвичай виконується перед більш детальними тестами.

10. Апаратне тестування: Це вид тестування, яке спрямоване на перевірку апаратних компонентів системи, таких як процесори, пам'ять, сенсори тощо. Воно включає тестування працездатності, швидкості та стабільності апаратних компонентів.

11. Відмовостійке тестування: Це вид тестування, яке спрямоване на виявлення помилок та відмов системи. Його метою є перевірка стійкості системи до несподіваних відмов та відновлення працездатності після відмови.

Ці види тестування використовуються для забезпечення якості та надійності системи під час її розробки та впровадження. Комбінація різних видів тестування дозволяє виявити широкий спектр проблем та забезпечити належну роботу системи (Додаток Г).

Для перевірки вірності роботи програмного забезпечення експертної системи ОТС використовувались декілька контрольних прикладів розрахунку різних ймовірностей подій. На рис. 3.7 наведений контрольний приклад розрахунку ймовірності спрацювання тривожної сигналізації. Як бачимо, результат, наведений на рис. 3.8 співпадає з результатом наданим експертною системою (див. рис. 3.7)

$$\begin{aligned}
 &P(a) \\
 &P(a) = P(a|s) * P(s) + P(a|\bar{s}) * P(\bar{s}) \\
 &P(s) = P(s|b,e) * P(b) * P(e) + P(s|b,\bar{e}) * P(b) * \\
 &P(\bar{e}) + P(s|\bar{b},e) * P(\bar{b}) * P(e) + P(s|\bar{b},\bar{e}) * \\
 &P(\bar{b}) * P(\bar{e}) = 0,9 * 0,001 * 0,02 + 0,9 * 0,001 * \\
 &0,98 + 0,1 * 0,999 * 0,02 + 0,001 * 0,999 * 0,98 \\
 &= 0,00387702 \\
 &P(\bar{s}) = 1 - P(s) = 1 - 0,00387702 = 0,99612298 \\
 &P(a) = 0,95 * 0,00387702 + 0,001 * 0,99612298 = \\
 &= 0,003683169 + 0,00099612298 = 0,0046792919
 \end{aligned}$$

де,

a — alarm

b — bulgary

s — sensor

Рис. 3.8. Розрахунок вірогідності спрацювання тривожної сигналізації.

ВИСНОВКИ

У даній дипломній роботі була розглянута тема розробки експертної системи з байєсівською системою логічного висновку. Досліджено теоретичні аспекти та проаналізовано стан розробок експертних систем для розв'язання інтелектуальних задач. Було визначено поняття, призначення та структуру експертної системи, а також порівняно різні моделі представлення знань.

У другому розділі дипломної роботи було проведено проектування експертної системи з байєсівською системою логічного висновку для охоронної тривожної сигналізації. Була сформульована постановка задачі та розроблена математична модель міркування в байєсівських мережах. Також був розроблений алгоритм розрахунку імовірнісних параметрів експертної системи.

У третьому розділі дипломної роботи було проведено розробку програмного забезпечення експертної системи з байєсівською системою логічного висновку. Була реалізована програмна реалізація експертної системи та виконано розробку, налагодження та виконання програмного коду. Були визначені компоненти системи, організована база знань та реалізовано механізм байєсівського логічного висновку.

- Розроблена експертна система з байєсівською системою логічного висновку показала свою ефективність у розв'язанні інтелектуальних задач, зокрема в охоронній тривожній сигналізації.
- Використання байєсівської системи логічного висновку дозволяє забезпечити точніші та обґрунтовані результати в порівнянні з іншими методами висновку.
- Розроблена програмна реалізація експертної системи забезпечує зручний та ефективний інтерфейс для користувача та може бути успішно використана в різних сферах, де потрібна підтримка прийняття рішень.

У ході виконання дипломної роботи на тему "Експертна система з байєсівською системою логічного висновку" були розглянуті основні принципи та концепції байєсівських мереж та їх застосування у експертних системах. Було

вивчено принципи роботи байєсівських мереж, включаючи визначення змінних, станів та ймовірнісних параметрів.

Було розглянуто процес створення бази знань та налагодження програмного коду для експертної системи з використанням пакету SWI-Prolog. Було описано загальні кроки для розрахунку в байєсівській мережі, включаючи визначення станів змінних, ймовірнісних параметрів та розрахунок умовних ймовірностей.

Для виконання дипломної роботи було використано різноманітні джерела, включаючи наукові статті, публікації конференцій, книги та веб-ресурси. Ці джерела надали необхідні теоретичні та практичні знання про байєсівські мережі та експертні системи. В результаті проведених досліджень було встановлено, що експертні системи з байєсівською системою логічного висновку є потужним інструментом для моделювання складних проблем та прийняття рішень в умовах невизначеності. Вони дозволяють поєднати експертні знання та ймовірнісний аналіз, що призводить до більш обґрунтованих та надійних висновків. У подальшому дослідженні можна розширити функціональні можливості експертної системи, додати нові правила та змінні, щоб забезпечити більш гнучкий та повний аналіз проблем. Також можна провести експерименти з різними методами вивчення параметрів байєсівської мережі для поліпшення точності та швидкодії системи. Отже, дипломна робота дала можливість вивчити теоретичні аспекти експертних систем з байєсівською системою логічного висновку, провести проектування та розробку програмного забезпечення такої системи. Результати роботи мають практичне значення і можуть бути використані для подальшого розвитку та застосування експертних систем в різних сферах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лакіза В. І., Войтович В. В., Коваленко М. В. Штучний інтелект: методи та засоби / В. І. Лакіза, В. В. Войтович, М. В. Коваленко, 2018. – 218 с.
2. Bishop C. Pattern Recognition and Machine Learning [Електроний ресурс]. – Режим доступу: <https://www.numerade.com/books/chapter/introduction-166/>
3. Koller D., N. Friedman Probabilistic Graphical Models: Principles and Techniques / D. Koller, F. Friedman – Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. 1988. – 97.
4. Байєсівська мережаю: [Електроний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Байєсівська_мережа
5. Барбер Д. Байєсівський висновок та машинне навчання / Д. Барбер – Cambridge University Press, 2012.
6. Барінов А. Н., Булатов А. А. Искусственный интеллект: модели и методы / А. Н. Барінов, А. А. Булатов – Київ: Діалог-МИФИ, 2012.
7. Биков Л.Б. Штучний інтелект: Навчальний посібник / Л.Б. Биков – Київ: Національний технічний університет України "Київський політехнічний інститут", 2014. – 379 с.
8. Блог "Machine Learning Mastery": [Електроний ресурс]. – Режим доступу: <https://machinelearningmastery.com/>
9. Бондаренко В. Експертні системи: теорія та практика / В. Бондаренко – Наукові праці Національного університету "Львівська політехніка". 2009
10. Васильєв В. Ю., Шматко А. М. Експертні системи на основі байєсівських мереж / В. Ю. Васильєв, А. М. Шматко. – Матеріали науково-практичної конференції "Сучасні проблеми теорії та практики інформатики" 2019. – 165 с.
11. DataCamp: [Електроний ресурс]. – Режим доступу: <https://www.datacamp.com/>
12. Українська Асоціація Штучного Інтелекту: [Електроний ресурс]. – Режим доступу: <http://uai.org.ua/>

13. Галушкін, А. І., Качан, О. М. Байєсівські мережі в інтелектуальних системах / А. І. Галушкін, О. М. Качан. – Вісник Житомирського державного технологічного університету. 2018. – 118 с.
14. Гринь, О. М., Білова, О. Л. Аналіз методів побудови байєсівських мереж в експертних системах / О. М. Гринь, О. Л. Білова. Наукові записки Національного університету "Острозька академія". Серія: Комп'ютерні системи та мережі. 2009 – 112 с.
15. Дарвіч, А. Моделювання та висновки за допомогою байєсівських мереж / А. Дарвіч. Cambridge University Press – 2009
16. Друждзель М. Й., Байєсівські мережі / М. Й. Друждзель. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2016 – 110 с.
17. Дубровіна, В. В., Молчанов, А. О., Системи штучного інтелекту: навчальний посібник / В. В. Дубровіна, А. О. Молчанов. Київ: Видавництво Національного авіаційного університету. 2019
18. Єнсен Ф. В., Нільсен Т. Д.. Байєсовські мережі та графи рішень / Ф. В. Єнсен, Т. Д. Нільсен. Springer Science & Business Media. 2007
19. Журавель О. В., Семенов І. М. Використання байєсівських мереж для побудови експертних систем / О. В. Журавель, І. М. Семенов. Вісник Черкаського державного технологічного університету. 2017 – 60 с.
20. Журавель О. М., Хоменко О. В. Експертні системи: підручник / О. М. Журавель, О. В. Хоменко. Київ: Видавничо-поліграфічний центр "Київський університет". 2014
21. Bayesian Networks and Decision Graphs: [Електроний ресурс]. – Режим доступу: [https://www.google.com.ua/books/edition/Bayesian_Networks_and_Decision_Graphs/37CAgCyxQaAC?hl=ru&gbpv=1&dq=inauthor:"Thomas+Dyhre+Nielsen"&printsec=frontcover](https://www.google.com.ua/books/edition/Bayesian_Networks_and_Decision_Graphs/37CAgCyxQaAC?hl=ru&gbpv=1&dq=inauthor:)
22. Костюк О. В., Руденко А. В. Байєсовські мережі в експертних системах / О. В. Костюк, А. В. Руденко. Вісник Київського національного університету імені Тараса Шевченка. 2016 – 84 с.

23. Кравченко В.А. Експертні системи: моделі та технології: Навчальний посібник / В.А. Кравченко. Київ: 2006. – 520 с.
24. Купер Г. Ф., Герсковіц Е. Байєсовський метод індукції ймовірнісних мереж на основі даних. Машинне навчання / Г. Ф. Купер, Е. Герсковіц. 1997. – 347 с.
25. Probabilistic Graphical Models: [Електронний ресурс]. – Режим доступу: <https://www.coursera.org/learn/probabilistic-graphical-models>
26. Куценко О. М. Байєсівські мережі / О. М. Куценко. Київ: Видавничий дім "Слово". 2011
27. Литвин І. В., Борщевський В. Г. . Методи і алгоритми побудови байєсівських мереж в експертних системах / І. В. Литвин, В. Г. Борщевський. Вісник Харківського національного університету Повітряних Сил імені Івана Кожедуба. 2021 – 55 с.
28. Маркова О. О. Використання байєсівських мереж для побудови експертних систем / О. О. Маркова. Науковий вісник НЛТУ України. 2019 – 175 с.
29. Матеріали конференції "Інтелектуальний аналіз даних" (ІАД): [Електронний ресурс]. – Режим доступу: <http://iad.icmp.lviv.ua/>
30. Медведєв В. В., Лаврова О. В. Використання експертних систем у фінансових організаціях / В. В. Медведєв, О. В. Лаврова. Вісник Національного технічного університету "Харківський політехнічний інститут". 2012.
31. Мельничук В. І., Синяк В. О. Застосування байєсівської мережі у задачі діагностики / В. І. Мельничук, В. О. Синяк. Науковий вісник Національного гірничого університету. 2018 – 61 с.
32. Наукові бази даних та пошукові системи: Google Scholar: [Електронний ресурс]. – Режим доступу <https://scholar.google.com/>); IEEE Xplore: [Електронний ресурс]. – Режим доступу <https://ieeexplore.ieee.org/>; ACM Digital Library: [Електронний ресурс]. – Режим доступу <https://dl.acm.org/>
33. Неаполітан Р. Є. Вивчення байєсівських мереж / Р. Є. Неаполітан. Prentice Hall. 2004

34. Перл Дж. Ймовірнісний висновок в інтелектуальних системах: мережі правдоподібних висновків / Дж. Перл. Morgan Kaufmann. 1988
35. Петренко І. О., Миколайчук О. В. Байесовські мережі в системах штучного інтелекту / І. О. Петренко, О. В. Миколайчук. Матеріали Всеукраїнської науково-практичної конференції "Інформаційні технології та комп'ютерне моделювання". 2018 – 241 с.
36. Попов І. Методи побудови байесівських мереж в експертних системах / І. Попов. Вісник Харківського національного університету імені В. Н. Каразіна. 2013 – 193 с.
37. Придатко В. І., Соломко В. П. Експертні системи та штучний інтелект / В. І. Придатко, В. П. Соломко. Київ: Видавничий дім "Слово". 2016
38. Рассел С., Норвіг П. Штучний інтелект: сучасний підхід / С. Рассел, П. Норвіг. Pearson. 2016
39. Репозиторій GitHub з відкритим кодом для байесівських мереж та експертних систем: [Електроний ресурс]. – Режим доступу <https://github.com/>
40. Статті та документація з байесівських мереж: [Електроний ресурс]. – Режим доступу: <https://mlcourse.ai/articles/topic2-part2-bayesian/>
41. Ткачук В. В., Волошин В. В. Застосування байесівської мережі в експертних системах / В. В. Ткачук, В. В. Волошин. Вісник Чернігівського національного технологічного університету, 2019 – 64 с.
42. Фрідман Н., Мерфі К., Рассел С. Вивчення структури динамічних ймовірнісних мереж / Н. Фрідман, К. Мерфі, С. Рассел . Збірка матеріалів Чотирнадцятого конференції з нечітким штучним інтелектом. 1998 – 147 с.
43. Хекерман Д. Навчання з використанням байесівських мереж: посібник / Д. Хекерман. Звіт Microsoft Research, MSR-TR-95-06.1998
44. Чікерінг Д. М. Оптимальна ідентифікація структури з використанням жадібного пошуку / Д. М. Чікерінг. Журнал машинного навчання. 2002 – 507 с.
45. Шевченко В. І., Кобець М. В. Застосування байесівських мереж у системах експертних рішень / В. І. Шевченко, М. В. Кобець. Матеріали


Всеукраїнської науково-практичної конференції "Інформаційні технології та комп'ютерна інженерія". 2020 – 75 с.

46. Bayesian Methods for Hackers: [Електроний ресурс]. – Режим доступу: <https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers>
47. Bayesian Network Repository: [Електроний ресурс]. – Режим доступу: <http://www.bnlearn.com/bnrepository/>
48. Cowell R.G., Dawid A.P., Lauritzen S.L., Spiegelhalter D.J. Probabilistic Networks and Expert Systems / R.G. Cowell, A.P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter. New York: Springer, 1999. – 388 p.
49. Darwiche A. Modeling and Reasoning with Bayesian Networks / A. Darwiche. New York: Cambridge University Press, 2009. – 554 p.
50. Druzdzel M.J., van der Gaag L.C. (eds.) Principles of Uncertainty / M.J. Druzdzel, L.C. van der Gaag. New York: Chapman & Hall/CRC, 2019. – 530 p.
51. Fenton N.E., Neil M. Risk Assessment and Decision Analysis with Bayesian Networks / N.E. Fenton, M. Neil. Boca Raton: CRC Press, 2019. – 422 p.
52. Friedman N., Koller D. Probabilistic Graphical Models: Principles and Techniques / N. Friedman, D. Koller. Cambridge, MA: MIT Press, 2009. – 1137 p.
53. Giarratano J.C., Riley G.D. Expert Systems: Principles and Programming / J.C. Giarratano, G.D. Riley. Boston: Course Technology, 2004. – 688 p.
54. Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference / J. Pearl. San Francisco: Morgan Kaufmann, 1988. – 552 p.
55. Russell S., Norvig P. Artificial Intelligence: A Modern Approach / S. Russell, P. Norvig. Upper Saddle River, NJ: Pearson, 2016. – 1152 p.
56. Towards Data Science: [Електроний ресурс]. – Режим доступу: <https://towardsdatascience.com/>

ДОДАТКИ

Додаток А

Мова програмування Python для розробки програмної реалізації експертної системи ОТС



PYTHON TRICK

Use **Map** function of Python

<u>Don't do like this</u>	<u>Do like this</u>
<pre>def square(num): return num**2 mynums = [1,2,3,4,5] for x in mynums: print(square(x))</pre>	<pre>def square(num): return num**2 mynums = [1,2,3,4,5] sqrnum = list(map(square,mynums)) print(sqrnum)</pre>
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

FUTURE VISION

Додаток Б

Мова програмування Java для розробки програмної реалізації експертної системи ОТС

What is

Java
ASSIGNMENT HELP

JAVA?

Java is a programming language and computing platform. Java released by James Gosling at Sun Microsystems in 1995 and later developed by Oracle Corporation.

Java Platform?

Java platform is a set of programs that help to develop and run applications. Programmers use Java programming language to write these program codes.

Java-Related Technologies

- Java Swing,
- Java Threads,
- Polymorphism,
- Java Applets,
- Flowcharts,
- J2EEE,
- JavaScript,
- XML, etc.

Dynamic

Java is a dynamic language. At runtime, the Java environment can extend itself by linking in classes.

Secure

Java is Secure Language. It has many features it enables to develop tamper-free and virus-free systems.

Portable

The code of Java, written on one computer system. It can run on another computer system.

Multithreading

In Java multithreading is a highlight feature. It permits synchronized execution of two or more fragments of a program.

Simple

Java is simple Language. It has no Complex features like a pointer, operator overloading, inheritance, etc.

Java Is A Platform Independent Language

The compiler translates source code to byte-code.

Java Is An Object-Oriented Language

The way of Organizing Programs as a collection of objects. It represents an instance of a class.

Features Of Java

Resolve Your Java Assignment?
Codeavail provide help with java Assignment as per your instruction.

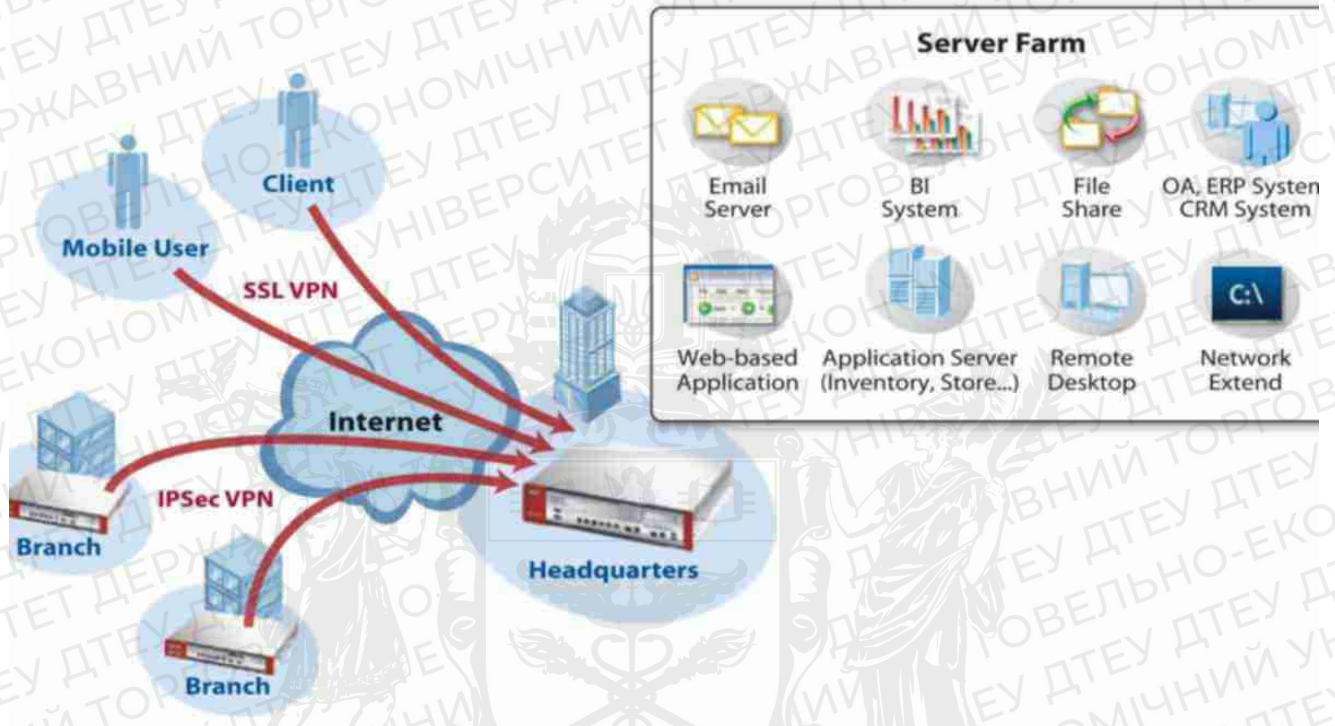
Need Java Assignment Help?
Students faces many financial problems and study time problems in their college life.

Java Bytecode
Programmers write Java programs in Java language. It compiles into Java bytecode.

Java Virtual Machine (JVM)
The Java Virtual Machine is like a computer system.

Додаток В

Байєсівські мережі: визначення, приклади та принципи роботи»



Додаток Г

Програмний код

`:- op(900, fx, not).`

`prob([X|Xs],Cond,P) :- !, % Вірогідність кон'юнкції`

`prob(X,Cond,Px),`

`prob(Xs,[X|Cond],Prest),`

`P is Px*Prest.`

`prob([],_,1) :- !. % Порожня кон'юнкція`

`prob(X,Cond,1) :-`
`my_member(X,Cond),!. % З умови Cond слідує X`

`prob(X,Cond,0) :-`
`my_member(not X,Cond), !. % З умови Cond слідує, що X - хибне`

`prob(not X,Cond,P) :- !,`

prob(X,Cond,P0),
P is 1 - P0.

% Застосувати закон Байєса, коли умова розповсюджується на дочірні вузли X

prob(X,Cond0,P) :-
my_delete(Y,Cond0,Cond),
predecessor(X,Y),!, % Y – дочірній вузол X
prob(X,Cond,Px),
prob(Y,[X|Cond],PyGivenX),
prob(Y,Cond,Py),
P is Px*PyGivenX / Py. % Передбачається, що $P_y > 0$

%Випадки, в яких умова не розповсюджується на дочірні вузли

prob(X,Cond,P) :-
p(X,P), !. % X- кореневий вузол; його ймовірність задана

prob(X,Cond,P) :- !,
findall((CONDi,Pi), p(X,CONDi,Pi), CPlist),
sum_probs(CPlist,Cond,P).

% sum_probs(CondsProbs,Cond,WeigthedSum):
% condsProbs - список умов та відповідних вірогідностей
% WeigthedSum) – зважена сума вірогідностей умов Conds
% при заданій умові Cond

sum_probs([], _,0).

sum_probs([(COND1,P1) | CondsProbs],COND,P) :-
prob(COND1,COND,PC1),
sum_probs(CondsProbs,COND,PRest),
P is P1*PC1+PRest.

predecessor(X, not Y) :-!, % Змінна Y, до якої задіяна
% операція заперечення

predecessor(X,Y).

predecessor(X,Y) :-
parent(X,Y).

predecessor(X,Z) :-
parent(X,Y),

predecessor(Y,Z).

my_member(X, [X | _]).

my_member(X,[_| L]) :-
my_member(X, L).

my_delete(X,[X|L],L).

my_delete(X, [Y|L],[Y|L2]) :-
my_delete(X,L,L2).

% Байєсівська мережа довіри

parent(burglary, sensor).
parent(lightning,sensor).
parent(sensor, alarm).
parent(sensor, call).

p(burglary, 0.001).
p(lightning, 0.02).
p(sensor, [burglary, lightning], 0.9).
p(sensor, [burglary, not lightning], 0.9).
p(sensor, [not burglary, lightning], 0.1).
p(sensor, [not burglary, not lightning], 0.001).
p(alarm, [sensor], 0.95).
p(alarm, [not sensor], 0.001).
p(call, [sensor], 0.9).
p(call, [not sensor], 0.0).

Додаток Д

Тестування програмного забезпечення: рівні, типи, етапи та методи налагодження

QA Process: Software Testing Stages

01 Requirement analysis.

To prevent missing or unclear essentials

03 Test case development.

To test individual code parts and features

05 Test execution.

To identify bugs and fix them

01

03

05

02

04

06

02 Test planning.

To pick tools and adopt a strategy

04 Testing environment set up.

To test the server, network, and devices

06 Cycle closure.

To evaluate QA project performance

