

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Людино-машинна взаємодія на основі байєсівського моделювання  
для інтеграції експертних знань та даних

Виконав(ла): студент(ка) 6 курсу, групи СНмн-61  
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Мага С.Ю.

(прізвище та ініціали)

Керівник

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Никитюк В.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.

(підпис)

(прізвище та ініціали)

"13" квітня 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр

(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки

(шифр і назва спеціальності)

студенту Мага Святослав Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи Людино-машинна взаємодія на основі байєсівського моделювання для інтеграції експертних знань та даних.

Керівник роботи д.т.н., проф. Литвиненко Я.В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від "10" березня 2026 року № 4/9-150

2. Термін подання студентом завершеної роботи 26 травня 2026 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити) ВСТУП; 1 ТЕОРЕТИЧНІ ЗАСАДИ; 1.1 Кореляція, асоціація та причинність; 1.2 Спрямовані ациклічні графи (DAG); 1.3 Теорема Байєса та імовірнісна теорія; 2 ЗНАННЯ ЕКСПЕРТА ЯК ДАНІ; 2.1 Формалізація знань; 2.2 Побудова моделі зі знань: випадок спринклера; 2.3 Умовно-ймовірнісні таблиці (СРТ); 2.4 Логічний висновок (інференс) на моделі; 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ; 3.1 Навчання структури (Structure Learning); 3.2 Навчання параметрів (Parameter Learning); 3.3 Інференс на основі даних; 3.4 Розширений приклад: передбачувальне обслуговування; 3.5 СРТ для системи технічного обслуговування; 3.6 Генерація синтетичних даних; 3.7 Бібліотека bnlearn; 3.8 Обмеження та практичні рекомендації; 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ; 4.1 Питання щодо охорони праці; 4.2 Питання щодо безпеки в надзвичайних ситуаціях; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ; ДОДАТКИ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Сенчишин В.С., к.т.н., доц. каф. МТ		
Безпека в надзвичайних ситуаціях	Теслюк В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 13 квітня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	13.04.26-18.04.26	Виконано
2.	Підбір наукових джерел за темою роботи	19.04.26-21.04.26	Виконано
3.	Переклад та опрацювання наукових джерел за темою кваліфікаційної роботи	20.04.26-23.04.26	Виконано
4.	Виконання дослідження щодо теми кваліфікаційної роботи	24.04.26-10.05.26	Виконано
5.	Оформлення першого розділу	04.05.26-05.05.26	Виконано
6.	Оформлення другого розділу	05.05.26-13.05.26	Виконано
7.	Оформлення третього розділу	13.05.26-14.05.26	Виконано
8.	Виконання завдання до підрозділу "Охорона праці"	08.05.26-09.05.26	Виконано
9.	Виконання завдання до підрозділу "Безпека в надзвичайних ситуаціях"	10.05.26-05.05.26	Виконано
10.	Оформлення кваліфікаційної роботи	05.05.26-13.05.26	Виконано
11.	Нормоконтроль	14.05.26-15.05.26	Виконано
12.	Перевірка на плагіат	16.05.2026	Виконано
13.	Попередній захист кваліфікаційної роботи	20.05.2026	Виконано
14.	Захист кваліфікаційної роботи	27.05.2026	

Студент

\_\_\_\_\_ (підпис)

Мага С.Ю.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Литвиненко Я.В.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

"Людино-машинна взаємодія на основі байєсівського моделювання для інтеграції експертних знань та даних" // Кваліфікаційна робота освітнього рівня "Магістр" // Мага Святослав Юрійович // Тернопільський національний технічний університет ім. І. Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2026 // с. – 52, рис. – 9, табл. – 2, джерел – 27.

Ключові слова: байєсівські мережі, причинно-наслідкове моделювання, спрямований ациклічний граф (DAG), умовно-ймовірнісні таблиці (CPT), навчання параметрів, логічний висновок, знання експерта, bnlearn, Python, синтетичні дані

Кваліфікаційна робота присвячена систематичному викладу концепцій та практичних методів роботи з байєсівськими мережами і причинно-наслідковим моделюванням засобами мови програмування Python із використанням відкритої бібліотеки bnlearn. Робота охоплює повний методологічний цикл: від формалізації якісних знань предметного експерта до автоматичного навчання структури й параметрів мережі з даних та здійснення логічного висновку (інференсу).

У теоретичній частині розмежовано поняття кореляції, асоціації та причинності, описано математичний апарат спрямованих ациклічних графів (DAG) і теорему Байєса. Розкрито підхід до трактування знань людини як повноцінного виду даних та продемонстровано їх формалізацію у вигляді змінних, зв'язків та умовно-ймовірнісних таблиць (CPT). На прикладі системи зрошення (Sprinkler) показано повний цикл побудови знання-керованої моделі, а на розширеному прикладі з галузі передбачувального технічного обслуговування – застосування байєсівських мереж за відсутності фізичних датчиків, включно з генерацією синтетичних даних. Окремо розглянуто методи

причинно-наслідкового структурного навчання (на основі оцінок та на основі обмежень) і навчання параметрів (метод максимальної правдоподібності та байєсівська оцінка).

Особливістю роботи є порівняльний аналіз результатів інференсу на моделі, побудованій зі знань експерта, та на моделі, навченій з емпіричних даних, що ілюструє природу суб'єктивних ймовірностей. Узагальнено функціональні можливості бібліотеки `bnlearn`, проаналізовано обмеження підходу (експоненційне зростання розмірності СРТ, обчислювальна складність пошуку оптимальної структури) та сформульовано практичні рекомендації.

## ANNOTATION

“Human-Computer Interaction Based on Bayesian Modeling for the Integration of Domain Knowledge and Data” // Master’s degree qualification paper // Maha Sviatoslav // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department, group CHHM-61 // Ternopil, 2026 // p. – 52, fig. – 9, tables – 2, references – 27.

Key words: bayesian networks, causal modeling, directed acyclic graph (DAG), conditional probability tables (CPT), parameter learning, logical inference, expert knowledge, bnlearn, Python, synthetic data

The qualification work is devoted to a systematic presentation of concepts and practical methods of working with Bayesian networks and causal modeling using the Python programming language using the open library bnlearn. The work covers the full methodological cycle: from formalization of qualitative knowledge of a subject matter expert to automatic learning of the structure and parameters of the network from data and implementation of logical conclusion (inference).

The theoretical part distinguishes the concepts of correlation, association and causality, describes the mathematical apparatus of directed acyclic graphs (DAG) and Bayes' theorem. An approach to treating human knowledge as a full-fledged type of data is disclosed and their formalization in the form of variables, relations and conditional probability tables (CPT) is demonstrated. The full cycle of knowledge-driven model construction is shown on the example of an irrigation system (Sprinkler), and the application of Bayesian networks in the absence of physical sensors, including the generation of synthetic data, is shown on an extended example from the field of predictive maintenance. The methods of causal structural learning (based on estimates and based on constraints) and parameter learning (maximum likelihood method and Bayesian estimation) are considered separately.

A feature of the work is a comparative analysis of the results of inference on a model built from expert knowledge and a model trained from empirical data, which illustrates the nature of subjective probabilities. The functional capabilities of the bnlearn library are summarized, the limitations of the approach are analyzed (exponential growth of the CPT dimension, computational complexity of finding the optimal structure) and practical recommendations are formulated.

## ЗМІСТ

ВСТУП.....	9
1 ТЕОРЕТИЧНІ ЗАСАДИ .....	16
1.1 Кореляція, асоціація та причинність .....	16
1.2 Спрямовані ациклічні графи (DAG).....	17
1.3 Теорема Байєса та імовірнісна теорія .....	18
2 ЗНАННЯ ЕКСПЕРТА ЯК ДАНІ.....	19
2.1 Формалізація знань .....	19
2.2 Побудова моделі зі знань: випадок спринклера.....	20
2.3 Умовно-ймовірнісні таблиці (CPT).....	21
2.4 Логічний висновок (інференс) на моделі.....	25
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ .....	26
3.1 Навчання структури (Structure Learning).....	26
3.2 Навчання параметрів (Parameter Learning).....	27
3.3 Інференс на основі даних .....	27
3.4 Розширений приклад: передбачувальне обслуговування .....	28
3.5 CPT для системи технічного обслуговування.....	29
3.6 Генерація синтетичних даних .....	29
3.7 Бібліотека bnlearn .....	30
3.8 Обмеження та практичні рекомендації.....	31
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ .....	32
4.1 Питання щодо охорони праці .....	32
4.1.1 Ергономіка .....	32
4.1.2 Освітлення .....	34
4.1.3 Параметри мікроклімату .....	35
4.1.4 Емоційна психогігієна .....	36
4.2 Питання щодо безпеки в надзвичайних ситуаціях .....	38
ВИСНОВКИ.....	40

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42
ДОДАТКИ.....	48

## ВСТУП

### **Актуальність задачі.**

У сучасних інформаційних системах обсяги даних зростають експоненційно, а методи машинного навчання стали стандартним інструментарієм для отримання інсайтів та побудови передбачень у різноманітних галузях – від медичної діагностики й рекомендаційних систем до промислового моніторингу та обробки природної мови. Попри суттєві досягнення у точності передбачень, домінуюча парадигма машинного навчання спирається майже виключно на дані: чим більший обсяг навчальної вибірки, тим надійніші результати. Однак ця парадигма має принципове обмеження, оскільки дані самі по собі не завжди відображають глибинні причинно-наслідкові механізми досліджуваної системи. Особливо гостро це проявляється в умовах, коли інформація є розрідженою, зашумленою, упередженою або взагалі недоступною – наприклад, за відсутності фізичних датчиків чи неможливості проведення контрольованих експериментів.

Саме в таких ситуаціях вирішальної ваги набувають знання предметних експертів. Проте людські знання здебільшого мають якісний, неоднозначний або неявний характер, що ускладнює їх безпосереднє опрацювання обчислювальними засобами. Виникає фундаментальна потреба у формальному апараті, здатному поєднати емпіричні спостереження з експертними судженнями в межах єдиної узгодженої моделі. Байєсівські мережі як клас імовірнісних графових моделей органічно вирішують це завдання, оскільки поєднують теорію графів із теорією ймовірностей і дозволяють як навчатися структурі залежностей безпосередньо з даних, так і кодувати апріорні знання експертів у вигляді спрямованих ациклічних графів та умовно-ймовірнісних таблиць.

Додаткової актуальності темі надає те, що в умовах поширення систем штучного інтелекту особливого значення набуває інтерпретованість моделей та можливість причинно-наслідкового міркування. На відміну від суто статистичних підходів, які виявляють лише кореляції та асоціації, байєсівські

мережі дозволяють робити обґрунтовані причинні висновки, моделювати наслідки втручань і відповідати на запитання типу «що відбудеться, якщо», що традиційно потребувало б проведення дороговартісних експериментів. Усе це зумовлює потребу в систематичному викладі концепцій та практичних прийомів роботи з байєсівськими мережами, доступному як для дослідників, так і для практиків.

### **Мета роботи.**

У сучасних інформаційних системах обсяги даних зростають експоненційно, а методи машинного навчання стали стандартним інструментарієм для отримання інсайтів та побудови передбачень у різноманітних галузях – від медичної діагностики й рекомендаційних систем до промислового моніторингу та обробки природної мови. Попри суттєві досягнення у точності передбачень, домінуюча парадигма машинного навчання спирається майже виключно на дані: чим більший обсяг навчальної вибірки, тим надійніші результати. Однак ця парадигма має принципове обмеження, оскільки дані самі по собі не завжди відображають глибинні причинно-наслідкові механізми досліджуваної системи. Особливо гостро це проявляється в умовах, коли інформація є розрідженою, зашумленою, упередженою або взагалі недоступною – наприклад, за відсутності фізичних датчиків чи неможливості проведення контрольованих експериментів.

Саме в таких ситуаціях вирішальної ваги набувають знання предметних експертів. Проте людські знання здебільшого мають якісний, неоднозначний або неявний характер, що ускладнює їх безпосереднє опрацювання обчислювальними засобами. Виникає фундаментальна потреба у формальному апараті, здатному поєднати емпіричні спостереження з експертними судженнями в межах єдиної узгодженої моделі. Байєсівські мережі як клас імовірнісних графових моделей органічно вирішують це завдання, оскільки поєднують теорію графів із теорією ймовірностей і дозволяють як навчатися структурі залежностей безпосередньо з даних, так і кодувати апіорні знання експертів у вигляді спрямованих ациклічних графів та умовно-ймовірнісних таблиць.

Додаткової актуальності темі надає те, що в умовах поширення систем штучного інтелекту особливого значення набуває інтерпретованість моделей та можливість причинно-наслідкового міркування. На відміну від суто статистичних підходів, які виявляють лише кореляції та асоціації, байєсівські мережі дозволяють робити обґрунтовані причинні висновки, моделювати наслідки втручань і відповідати на запитання типу «що відбудеться, якщо», що традиційно потребувало б проведення дороговартісних експериментів. Усе це зумовлює потребу в систематичному викладі концепцій та практичних прийомів роботи з байєсівськими мережами, доступному як для дослідників, так і для практиків.

Для того, щоби досягти

Для досягнення поставленої мети в роботі сформульовано та розв'язано такі основні завдання:

1) розглянути та систематизувати теоретичні засади причинно-наслідкового моделювання, зокрема розмежувати поняття кореляції, асоціації та причинності, а також описати математичний апарат спрямованих ациклічних графів і теореми Байєса;

2) дослідити підхід до трактування знань експерта як повноцінного виду даних та описати процес їх формалізації у вигляді змінних, зв'язків та ймовірностей;

3) продемонструвати на прикладі системи зрошення (Sprinkler) повний цикл побудови знання-керованої моделі – від формулювання пропозицій експерта до визначення умовно-ймовірнісних таблиць та здійснення логічного висновку;

4) проаналізувати методи причинно-наслідкового структурного навчання, зокрема підходи на основі оцінок та на основі обмежень, а також методи навчання параметрів мережі з даних;

5) розробити розширений практичний приклад із галузі передбачувального технічного обслуговування, що включає побудову DAG, визначення СРТ та генерацію синтетичних даних;

б) узагальнити функціональні можливості бібліотеки `bnlearn` та сформулювати практичні рекомендації щодо подолання обмежень підходу.

У сучасному світі, де обробка даних є ключовим чинником прийняття рішень, дедалі більшого значення набувають методи, що дозволяють поєднувати емпіричні спостереження з формалізованими знаннями предметних експертів. Традиційні підходи машинного навчання орієнтовані виключно на дані: чим більший обсяг навчальної вибірки, тим точніші передбачення. Однак ця парадигма має суттєве обмеження – дані самі по собі не завжди відображають глибинні причинно-наслідкові механізми досліджуваної системи, особливо в умовах браку спостережень, зашумлених або неповних даних.

Байєсівські мережі (Bayesian Networks, BN) являють собою потужний клас імовірнісних графових моделей, що органічно поєднують теорію графів із теорією ймовірностей. Ці моделі дозволяють як навчатися структурі залежностей з даних (причинно-наслідкове структурне навчання), так і кодувати апріорні знання експертів у вигляді спрямованих ациклічних графів (Directed Acyclic Graphs, DAG) та умовно-ймовірнісних таблиць (Conditional Probability Tables, CPT).

Метою цієї роботи є систематичний виклад ключових концепцій та практичних прийомів роботи з байєсівськими мережами засобами мови програмування Python із використанням бібліотеки `bnlearn`.

Концептуальним стрижнем, що об'єднує всі розділи роботи, є ідея людино-машинної співпраці (human-machine collaboration). Йдеться не про заміну людського судження алгоритмами і не про сліпе довіряння даним, а про продуктивне партнерство, у якому людина-експерт формулює структурні припущення та причинні гіпотези про систему, а обчислювальна модель надає математично узгоджений апарат для їх перевірки, поєднання з емпіричними спостереженнями та кількісного виведення наслідків. Байєсівські мережі є чи не найприроднішим середовищем для такої співпраці, оскільки вони здатні однаково сприймати як «м'які» знання людини, виражені у вигляді переконань і евристик, так і «жорсткі» статистичні закономірності, відновлені з даних.

Структурно робота складається з трьох розділів. Перший розділ присвячено теоретичному підґрунтю: розмежуванню понять кореляції, асоціації та причинності, а також опису математичного апарату DAG та теореми Байєса. Другий розділ демонструє, як знання експерта можуть бути трансформовані у формалізовану байєсівську модель. Третій розділ висвітлює методи автоматичного навчання структури та параметрів мережі з даних а містить розширений практичний приклад із галузі передбачувального технічного обслуговування. Також тут описано можливості бібліотеки `bnlearn` та узагальнено обмеження підходу з формулюваннями практичних рекомендацій.

Весь програмний код, що ілюструє матеріал посібника, зібрано в Додатку А із відповідними посиланнями у тексті.

**Об’єкт дослідження:** процес людино-машинної взаємодії у побудові імовірнісних моделей, тобто поєднання знань предметного експерта з емпіричними даними засобами байєсівського моделювання.

**Предмет дослідження:** методи, моделі та програмні засоби формалізації знань експерта й причинно-наслідкового навчання за допомогою байєсівських мереж (зокрема спрямовані ациклічні графи, умовно-ймовірнісні таблиці, алгоритми структурного навчання й навчання параметрів та процедури логічного висновку), реалізовані в бібліотеці `bnlearn`.

### **Наукова новизна отриманих результатів.**

Наукова новизна роботи полягає в системному поєднанні двох взаємодоповнювальних підходів до байєсівського моделювання, які зазвичай розглядаються окремо: знання-керованого моделювання, що ґрунтується на апіорних судженнях експерта, та даних-керованого структурного навчання, що відновлює причинно-наслідкову структуру безпосередньо зі спостережень. У роботі вперше для україномовного навчального контексту цілісно викладено повний методологічний цикл – від формалізації якісних знань експерта до автоматичного навчання структури й параметрів мережі та здійснення інференсу – на основі єдиного програмного інструментарію.

Новим елементом є також порівняльний аналіз результатів інференсу, отриманих на моделі, побудованій виключно зі знань експерта, та на моделі, навченій з емпіричних даних. Таке зіставлення наочно демонструє природу суб'єктивних ймовірностей та пояснює розходження між експертними оцінками й статистичними розподілами, що має методологічну цінність для розуміння меж застосовності кожного з підходів. Крім того, у роботі систематизовано критичний аналіз обмежень байєсівських мереж, зокрема експоненційного зростання розмірності умовно-ймовірнісних таблиць та обчислювальної складності пошуку оптимальної структури, що дозволяє сформулювати обґрунтовані рекомендації щодо вибору методів залежно від масштабу задачі.

### **Практичне значення отриманих результатів.**

Практична цінність дослідження визначається передусім тим, що весь викладений матеріал супроводжується робочим програмним кодом мовою Python із використанням відкритої бібліотеки `bnlearn`, зібраним у структурований додаток із посиланнями на відповідні розділи тексту. Це дозволяє читачеві не лише засвоїти теоретичні концепції, а й безпосередньо відтворити всі приклади, адаптувати їх під власні задачі та інтегрувати у власні проєкти. Покроковий характер викладу робить роботу придатною для використання як навчального посібника під час підготовки фахівців із аналізу даних та машинного навчання.

Запропонований підхід має пряме прикладне застосування в галузях, де якісні експертні знання традиційно відігравали провідну роль, а збір повноцінних даних є ускладненим або надто витратним – зокрема в медицині, інженерії, оцінюванні ризиків та промислового моніторингу. Розширений приклад із передбачувального технічного обслуговування демонструє, як за відсутності фізичних датчиків можна побудувати працездатну ймовірнісну модель відмов обладнання виключно на основі знань досвідченого спеціаліста, а згодом використати її для генерації синтетичних даних. Нарешті, сформульовані практичні рекомендації щодо організації інтерв'ю з експертами, узгодження вербальних формулювань ймовірностей та проведення перевірок розумності

моделі мають безпосередню операційну цінність для команд, що впроваджують байєсівські методи у реальні робочі процеси.

**Апробація результатів та особистий внесок здобувача.**

Основні положення роботи доповідались, розглядались та обговорювались на науковій конференції Тернопільського національного технічного університету імені Івана Пулюя у тезах студентської науково-технічної конференції "Природничі та гуманітарні науки. Актуальні питання – 2026", яка проходила у ТНТУ.

## 1 ТЕОРЕТИЧНІ ЗАСАДИ

### 1.1 Кореляція, асоціація та причинність

Використання методів машинного навчання стало стандартним інструментарієм для отримання корисних інсайтів і побудови передбачень у різних галузях – від прогнозування захворювань до рекомендаційних систем та обробки природної мови. Однак, незважаючи на досягнення у якості передбачень, виявлення причинно-наслідкових зв'язків залишається нетривіальним завданням. Ключовим питанням є таке: які змінні мають прямий причинний вплив на цільову змінну?

Необхідно чітко розмежувати три поняття, що часто вживаються як синоніми, але мають принципово різне значення: кореляція, асоціація та причинність.

Кореляція Пірсона є найпоширенішим коефіцієнтом кореляції, що вимірює силу лінійного зв'язку між двома змінними за шкалою від  $-1$  до  $1$ . Розрізняють позитивну, негативну та нульову кореляцію. Показовим прикладом хибної кореляції є виявлений зв'язок між споживанням шоколаду та кількістю лауреатів Нобелівської премії за країнами – очевидно, між цими явищами немає причинно-наслідкового зв'язку.

Асоціація у статистичному сенсі означає, що певні значення однієї змінної мають тенденцію до спільного виникнення з певними значеннями іншої. Для її вимірювання застосовуються тест  $\chi^2$ -квадрат, точний тест Фішера, гіпергеометричний тест та інші. Асоціація використовується, коли одна або обидві змінні є категоріальними.

Причинність означає, що одна (незалежна) змінна зумовлює зміну іншої (залежної) змінної. Відповідно до формулювання Reichenbach (1956): якщо дві випадкові змінні  $X$  та  $Y$  є статистично залежними ( $X/Y$ ), то або  $X$  є причиною  $Y$ , або  $Y$  є причиною  $X$ , або існує третя змінна  $Z$ , що є причиною і  $X$ , і  $Y$ ; при цьому

$X$  та  $Y$  є незалежними при умові  $Z$ , тобто  $X \perp Y | Z$ . Це визначення безпосередньо інкорпороване у байєсівські графові моделі.

Для ілюстрації використання асоціативного аналізу розглянемо набір даних про передбачувальне технічне обслуговування (см. Лістинг А.1 у Додатку А). Гіпергеометричний тест дозволяє перевірити нульову гіпотезу про відсутність асоціації між відмовою машини та відмовою через перевантаження (PWF). При  $M = 10000$  (розмір вибірки),  $n = 339$  (кількість відмов машини),  $N = 95$  (кількість подій PWF) та  $x = 95$  (перетин),  $p$ -значення становить  $1.66 \times 10^{-146}$ , що дозволяє відхилити нульову гіпотезу. Однак асоціація сама по собі не імплікує причинності.

## 1.2 Спрямовані ациклічні графи (DAG)

Байєсівська мережа є спрямованим ациклічним графом (Directed Acyclic Graph, DAG) – мережею (або графом) із вузлами (змінними) та ребрами, що мають єдиний напрямок (див. рис. 1.1). Вузли відповідають змінним, а спрямовані ребра (стрілки) вказують на відношення умовної залежності між парами змінних. Мережа є ациклічною, тобто не допускає зворотних (фідбекових) петель.

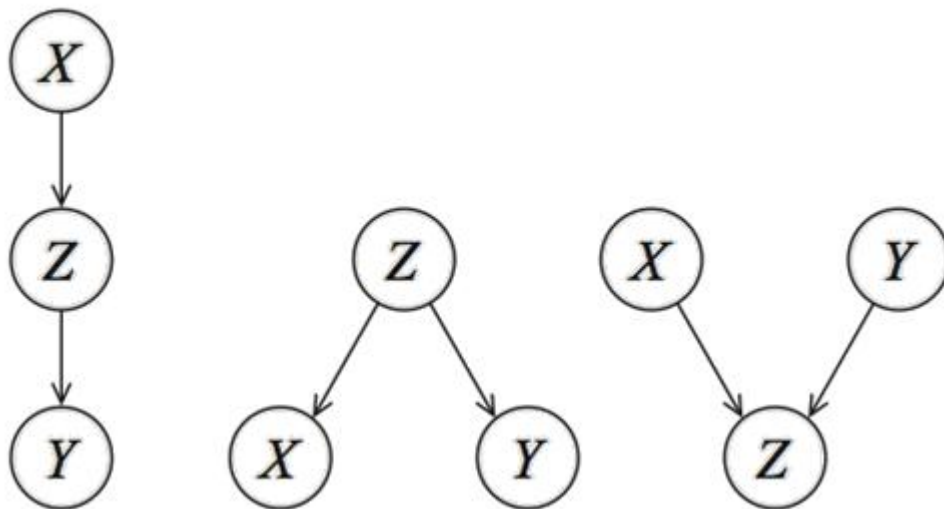


Рисунок 1.1 – Унікальні патерни DAG для трьох вузлів

Усі DAG (великі чи малі) будуються за трьома правилами:

- ребра є умовними залежностями;
- ребра мають напрямок;
- зворотні петлі заборонені.

Для трьох змінних ( $X$ ,  $Y$ ,  $Z$ ) існують чотири унікальних патерни: два каскадних (a та b), один – зі спільним батьківським вузлом (c) та V-структура (d). Ці чотири граfi є основою кожної байєсівської мережі. Важливо, що без урахування напрямку ребер перші три граfi виглядали б ідентично – саме спрямованість робить DAG ідентифікованим.

Концептуальна ідея визначення напрямку причинності полягає в утриманні одного вузла сталим і спостереженні за ефектом. Наприклад, для DAG (a), де  $Z$  спричинений  $X$ , а  $Y$  – спричинений  $Z$ : якщо утримати  $Z$  сталим, зміна  $X$  не повинна призводити до зміни  $Y$ , якщо модель коректна.

### 1.3 Теорема Байєса та імовірнісна теорія

Теорема Байєса (правило Байєса) формує математичний фундамент байєсівських мереж і застосовується для оновлення модельної інформації. Рівняння складається з чотирьох компонентів:

- апостеріорна ймовірність (posterior probability) – ймовірність того, що  $Z$  відбувається за умови  $X$ ;
- умовна ймовірність або правдоподібність (likelihood) – ймовірність отримати наявне свідчення за умови, що гіпотеза є істинною;
- апріорна ймовірність (prior) – ймовірність гіпотези до спостереження свідчень;
- маргінальна ймовірність (marginal probability) – ймовірність нового свідчення за всіма можливими гіпотезами.

Байєсівська мережа є щасливим шлюбом між теорією ймовірностей та теорією графів. Спільна розподіл для байєсівської мережі є добутком умовних ймовірностей для кожного вузла за умови його батьківських вузлів.

## 2 ЗНАННЯ ЕКСПЕРТА ЯК ДАНІ

### 2.1 Формалізація знань

У сучасному світі, де акцент часто робиться на зборі великих обсягів даних, знання експертів нерідко залишаються поза увагою. Проте дані самі по собі не завжди відображають глибинні механізми системи – особливо коли інформація є розрідженою, зашумленою, упередженою або просто недоступною. У таких випадках знання експертів відіграють вирішальну роль.

Знання людини – це також дані, хоча й такі, що зберігаються у свідомості людини, а не у базі даних. Ключова ідея моделювання, керованого знаннями, полягає в тому, що будь-яке експертне судження, причинне твердження або евристичне правило є фрагментом інформації про світ, а отже, його можна моделювати, квантифікувати і використовувати обчислювально.

Першим кроком перетворення знань на дані є формалізація того, що експерти знають про відношення між змінними: які чинники впливають на інші? Яка сила зв'язків? За яких умов вони виявляються? Наприклад, лікар може міркувати так: «певний симптом зазвичай виникає після конкретної інфекції». Це не рядок у таблиці, але структуровані спостереження щодо залежностей – той самий вид інформації, який статистичні моделі навчаються з даних, лише виражений вербально або концептуально.

Щоб зробити ці знання корисними для машин, їх необхідно закодувати в структуровану форму – у вигляді змінних, зв'язків і ймовірностей. Така репрезентація служить мостом між людським міркуванням і машинними обчисленнями.

Перетворення знань на дані відбувається за трьома основними кроками (рис. 1): (1) збір знань у вигляді евристик, правил, досвіду та припущень (рожевий крок); (2) структурування інформації у вигляді змінних, відношень та ймовірностей (синій крок); (3) представлення системи у вигляді графа з вузлами та ребрами (сірий крок) (див. рис. 2.1).

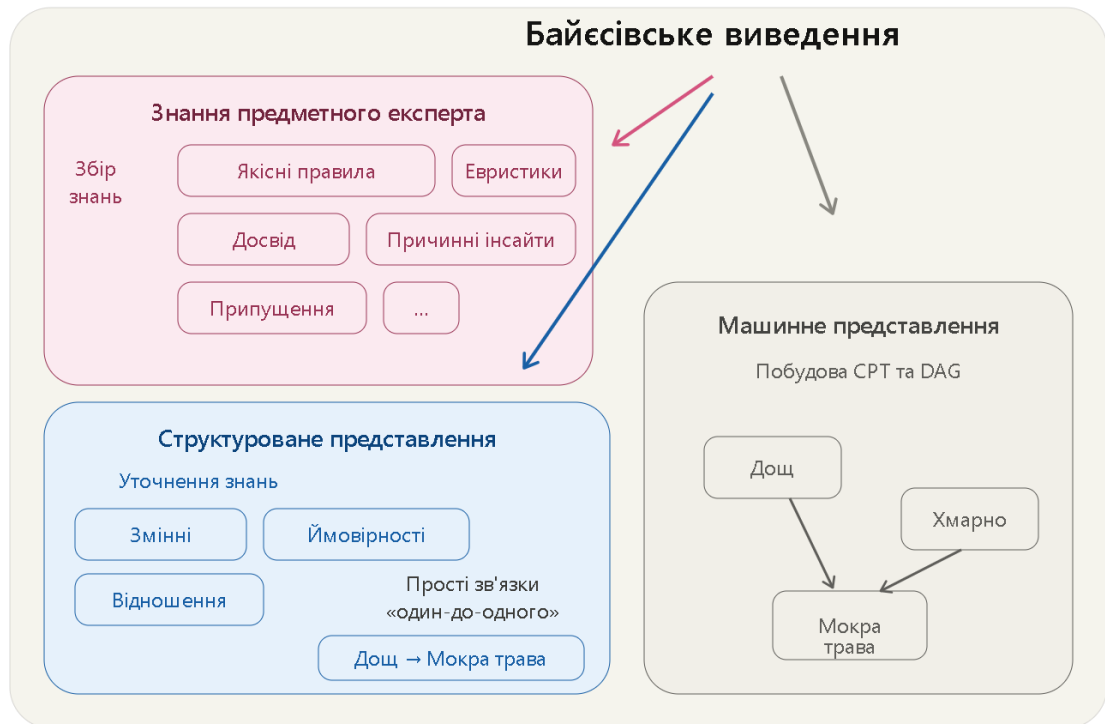


Рисунок 2.1 – Основні етапи моделювання знань експерта

Ефективність знання-керованого моделювання залежить від двох ключових аспектів: представлення знань експерта у вигляді графової структури та їх узгодженого об'єднання з використанням принципів теорії ймовірностей.

## 2.2 Побудова моделі зі знань: випадок спринклера

Розглянемо простий і наочний приклад побудови реальної моделі на основі знань експерта – систему зрошення газону (Sprinkler system).

Припустимо, що упродовж 1000 днів спостережень сформувався такі твердження (пропозиції) щодо системи:

- система зрошення іноді увімкнена, іноді вимкнена;
- якщо спринклер увімкнений, трава дуже часто є мокрою;
- дощ майже напевно призводить до мокрої трави, а спринклер найчастіше вимкнений;
- хмарна погода передуює дощу;
- існує слабкий зв'язок між спринклером та хмарністю.

На основі цих пропозицій визначаємо чотири вузли (кожен із двома станами) та чотири спрямовані однозв'язкові відношення:

- Хмарно → Дощ;
- Дощ → Мокра трава;
- Спринклер → Мокра трава;
- Хмарно → Спринклер.

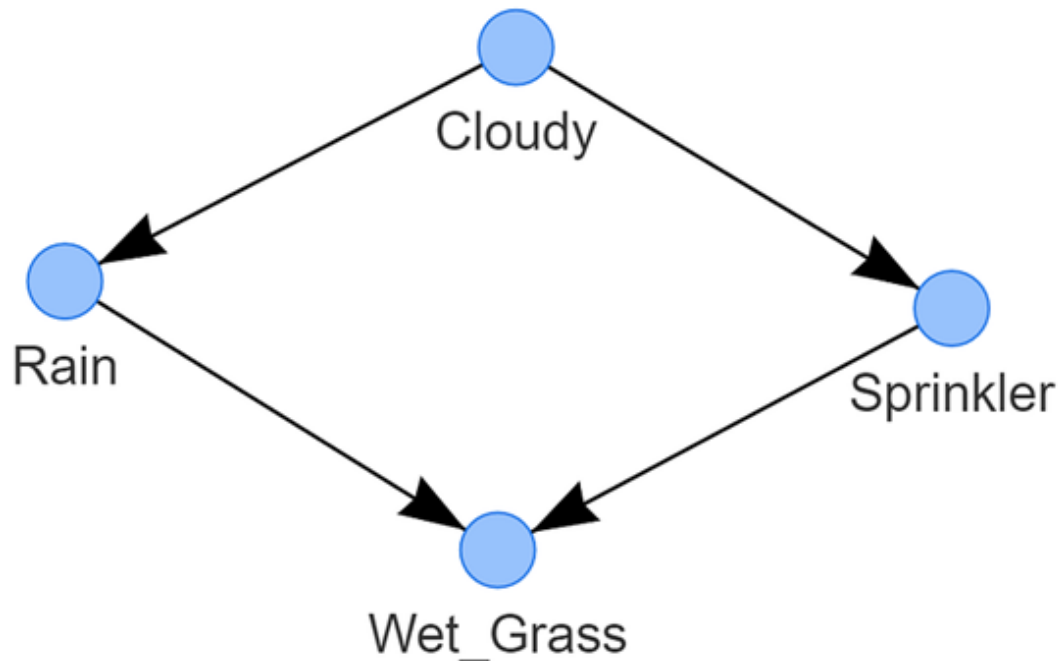


Рисунок 2.2 – DAG системи зрошення (Sprinkler)

Побудова DAG у середовищі Python з використанням бібліотеки `bnlearn` наведена у Лістингу A.2 (Додаток A). Результуючий граф показує: стан мокрої трави залежить від спринклера і дощу; дощ і стан спринклера залежать від хмарності.

### 2.3 Умовно-ймовірнісні таблиці (CPT)

CPT описують силу зв'язків між вузлами через умовні ймовірності. Для системи зрошення визначаємо чотири таблиці:

Вузел «Хмарно» не має батьківських залежностей. Виходячи з 70 % хмарних днів за 1000 спостережень:  $P(\text{Хмарно}=1)=0,70$ ;  $P(\text{Хмарно}=0)=0,30$  (Лістинг А.3) з виводом на рисунку 2.3.

```

+-----+-----+
| Cloudy(0) | 0.3 |
+-----+-----+
| Cloudy(1) | 0.7 |
+-----+-----+

```

Рисунок 2.3 – Вивід лістингу А.3

Вузел «Дощ» залежить від «Хмарно». За знаннями експерта: коли йшов дощ, 80% часу було хмарно; 20% дощів відбувалося без видимих хмар (Лістинг А.4) з виводом на рисунку 2.4.

```

+-----+-----+-----+
| Cloudy | Cloudy(0) | Cloudy(1) |
+-----+-----+-----+
| Rain(0) | 0.8 | 0.2 |
+-----+-----+-----+
| Rain(1) | 0.2 | 0.8 |
+-----+-----+-----+

```

Рисунок 2.4 – Вивід лістингу А.4

Вузел «Спринклер» залежить від «Хмарно». Коли спринклер був вимкнений, 90% часу було хмарно. Інші ймовірності встановлено на рівні 50% через невпевненість експерта (Лістинг А.5) з виводом на рисунку 2.5.

Cloudy	Cloudy(0)	Cloudy(1)
Sprinkler(0)	0.5	0.9
Sprinkler(1)	0.5	0.1

Рисунок 2.5 – Вивід лістингу А.5

Вузол «Мокра трава» залежить від «Дощу» та «Спринклера». Визначено вісім умовних ймовірностей:  $P(\text{Мокра}=1|\text{Дощ}=1, \text{Спр}=1)=0,99$ ;  $P(\text{Мокра}=0|\text{Дощ}=0, \text{Спр}=0)=1,0$ ;  $P(\text{Мокра}=1|\text{Дощ}=1, \text{Спр}=0)=0,90$  та ін. (Лістинг А.6).

Після визначення всіх таблиць DAG об'єднується з CPT та верифікується за допомогою функції `bn.print_CPD()`. Повний код побудови та перевірки наведено у Лістингу А.7 (див. рис. 2.6).

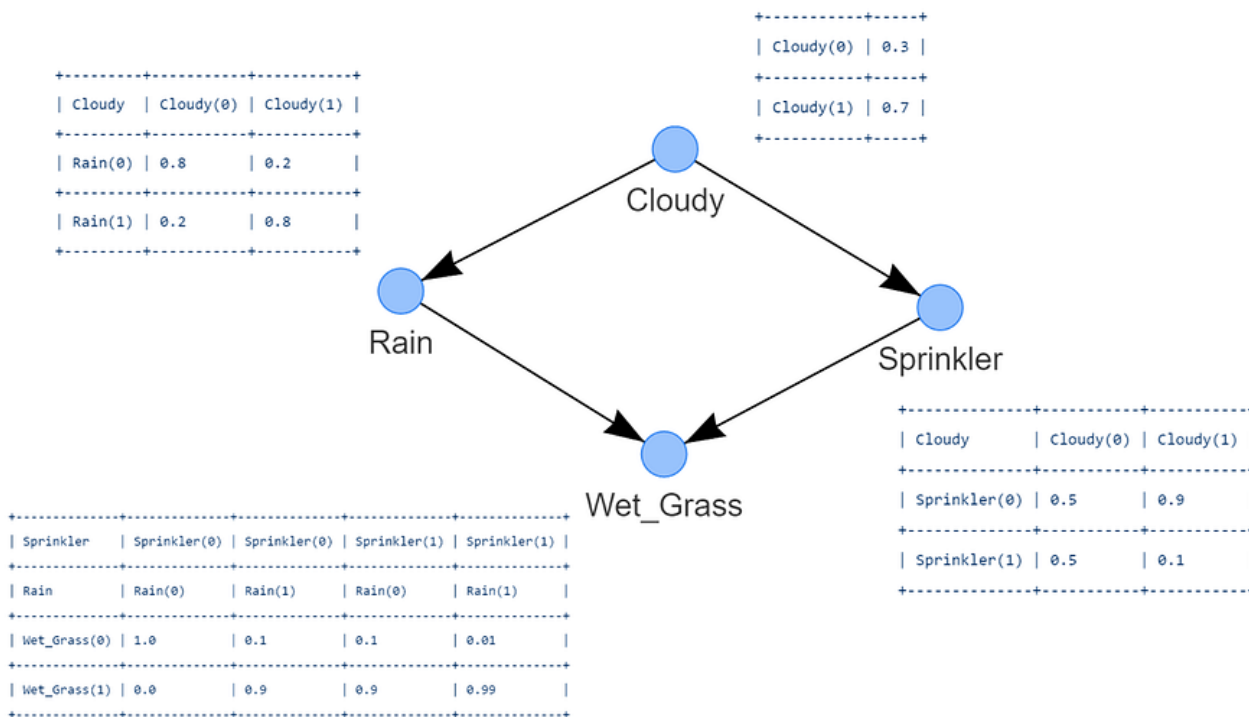


Рисунок 2.6 – Причинний DAG системи зрошення з приєднаними CPT

Слід зазначити, що ймовірності, одержані від експерта, є суб'єктивними ймовірностями (див. рис. 2.7).

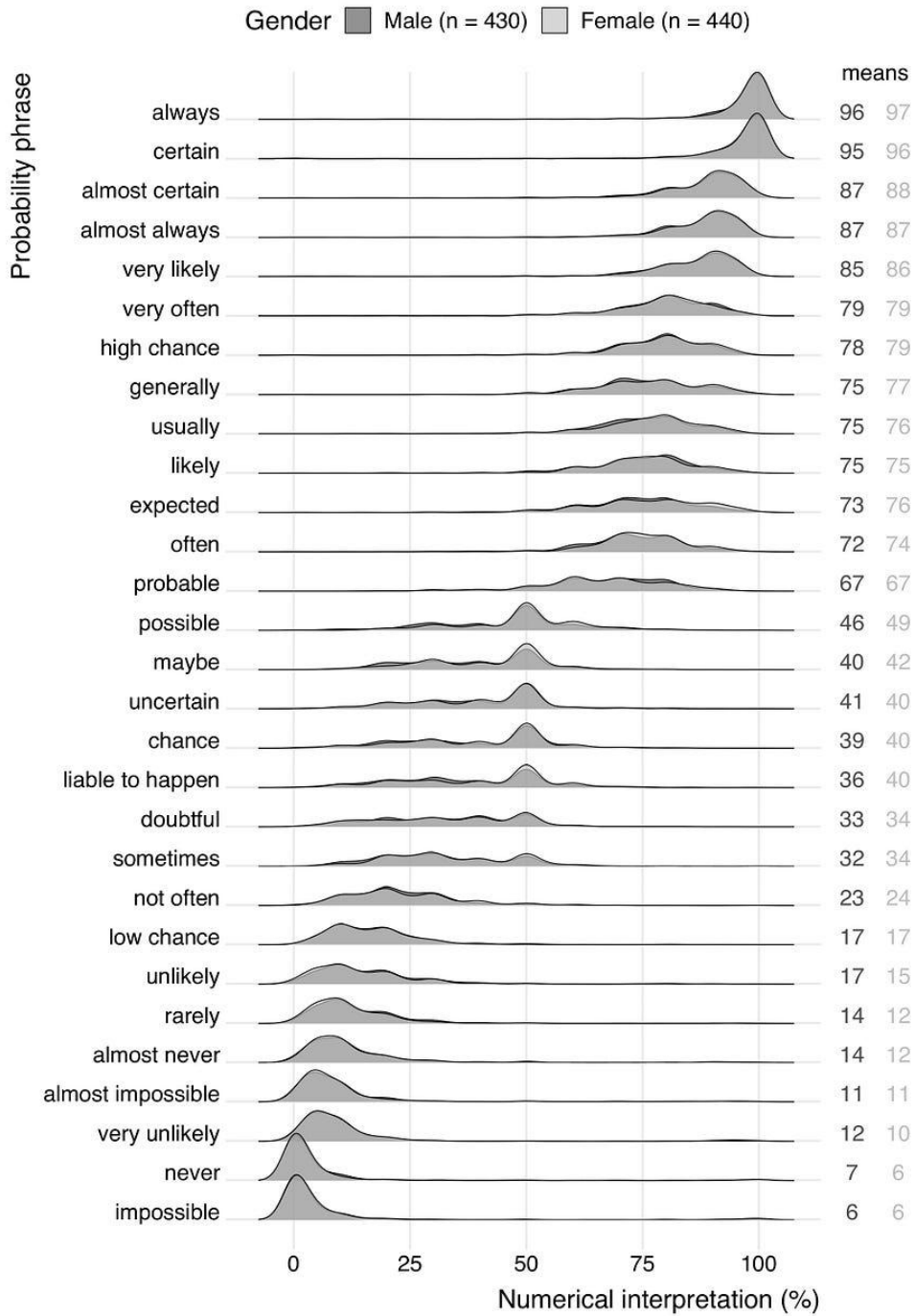


Рисунок 2.7 – Інтерпретація імовірностей, описаних вербально [4]

Вони відображають ступінь переконаності конкретної особи в конкретному місці та в конкретний час.

## 2.4 Логічний висновок (інференс) на моделі

На основі побудованої моделі можна здійснювати інференс – обчислення ймовірностей за заданими свідченнями. Приклади запитів та відповідні результати наведено у Лістингу А.8:

–  $P(\text{Мокра трава}=1 \mid \text{Спринклер}=0) \approx 0,616$  – якщо спринклер вимкнений, трава буде мокрою з ймовірністю 61,6 %;

–  $P(\text{Дощ}=1 \mid \text{Спринклер}=0, \text{Хмарно}=1) = 0,800$  – за вимкненого спринклера і хмарної погоди дощ очікується з ймовірністю 80 %;

–  $P(\text{Мокра трава}=1, \text{Дощ}=1 \mid \text{Спринклер}=1) \approx 0,387$  – одночасна ймовірність мокрої трави і дощу за увімкненого спринклера.

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ

### 3.1 Навчання структури (Structure Learning)

Структурне навчання (Structure Learning) має на меті визначення графа, що найкраще відображає причинно-наслідкові залежності між змінними у наборі даних. Наївний підхід передбачає перебір усіх можливих комбінацій графів і вибір найкращого за деяким критерієм якості. Однак простір пошуку DAG є над-експоненціальним за кількістю змінних, тому повний перебір практично нездійсненний для великих мереж. Натомість використовуються жадібні стратегії.

Існують два широкі підходи до структурного навчання:

- навчання на основі оцінок (Score-based Structure Learning);
- навчання на основі обмежень (Constraint-based Structure Learning).

Навчання на основі оцінок включає алгоритм пошуку (ExhaustiveSearch, HillClimbSearch, Chow-Liu, TAN) та функцію оцінки (BIC, K2, BDeu). HillClimbSearch реалізує жадібний локальний пошук, що починається з роз'єданого DAG і ітеративно виконує одиничні маніпуляції з ребрами, максимізуючи оцінку. Алгоритм Chow-Liu будує деревоподібну структуру, де кожен вузол має щонайбільше одного батька.

Навчання на основі обмежень використовує тести незалежності (тест хі-квадрат) для побудови DAG шляхом виявлення незалежностей у наборі даних. Прикладом є PC-алгоритм, що починає з повністю з'єданого графа та видаляє ребра на основі результатів тестів незалежності.

Практичний приклад структурного навчання для набору даних «Спринклер» (Лістинг А.9, Додаток А) демонструє, що налаштування bnlearn за замовчуванням використовує HillClimbSearch із оцінкою BIC. Виявлений DAG відображає такі залежності: ймовірність мокрої трави залежить від спринклера і дощу; ймовірність увімкнення спринклера залежить від хмарності; ймовірність дощу залежить від хмарності.

### 3.2 Навчання параметрів (Parameter Learning)

Навчання параметрів полягає в оцінці значень СРТ на основі наявних даних. Бібліотека `bnlearn` підтримує два основні методи:

Метод максимальної правдоподібності (Maximum Likelihood Estimation, MLE) використовує відносні частоти, з якими стани змінних спостерігалися у даних. Недоліком є схильність до перенавчання за малих вибірок – якщо у змінної є три батьківських вузли з десятьма станами кожен, оцінки стану обчислюватимуться для  $10^3 = 1000$  конфігурацій батьків, що може призвести до нестабільних оцінок.

Байєсівська оцінка (Bayesian Estimation) починається з готових апіорних СРТ, що відображають наші переконання до спостереження даних. Ці «пріори» оновлюються на основі підрахунку станів у спостережених даних. Простим пріором є  $K^2$  (додає «1» до підрахунку кожного стану); більш обґрунтованим вибором є  $BD_{eu}$  (рівномірний пріор Байєсівського Діріхле).

Приклад навчання параметрів для набору даних «Спринклер» із ручним обчисленням MLE наведено у Лістингах А.10 та А.11 (Додаток А).

### 3.3 Інференс на основі даних

Після отримання DAG і СРТ на основі даних можна здійснювати інференс. Метод усунення змінних (variable elimination) є алгоритмом точного інференсу; для великих мереж використовуються наближені алгоритми (вибірка Гіббса, відбракувальна вибірка).

Порівняно з моделлю, побудованою на основі знань експерта, модель, навчена з даних, дає дещо відмінні ймовірності. Зокрема:  $P(\text{Мокра трава}=1 \mid \text{Спринклер}=0) \approx 0,513$ ;  $P(\text{Дощ}=1 \mid \text{Спринклер}=0, \text{Хмарно}=1) \approx 0,663$  (Лістинг А.12, Додаток А). Ці розходження зумовлені тим, що навчена модель спирається на реальні розподіли у даних, тоді як модель на основі знань відображає суб'єктивні оцінки експерта.

### 3.4 Розширений приклад: передбачувальне обслуговування

Для демонстрації застосування байєсівських мереж у реалістичному промисловому контексті розглянемо задачу моделювання відмов обладнання. Припустимо, що фізичних датчиків для вимірювання параметрів немає, проте є досвідчений експерт, який надає таку інформацію:

«Відмови машин переважно спостерігаються за високої температури процесу або за високого крутного моменту. Високий крутний момент або знос інструмента призводить до відмов через перевантаження (OSF). На температуру процесу впливає температура навколишнього середовища.»

На основі цих відомостей формуються шість спрямованих однозв'язкових відношень (див. рис. 3.1):

- Температура процесу → Відмова машини;

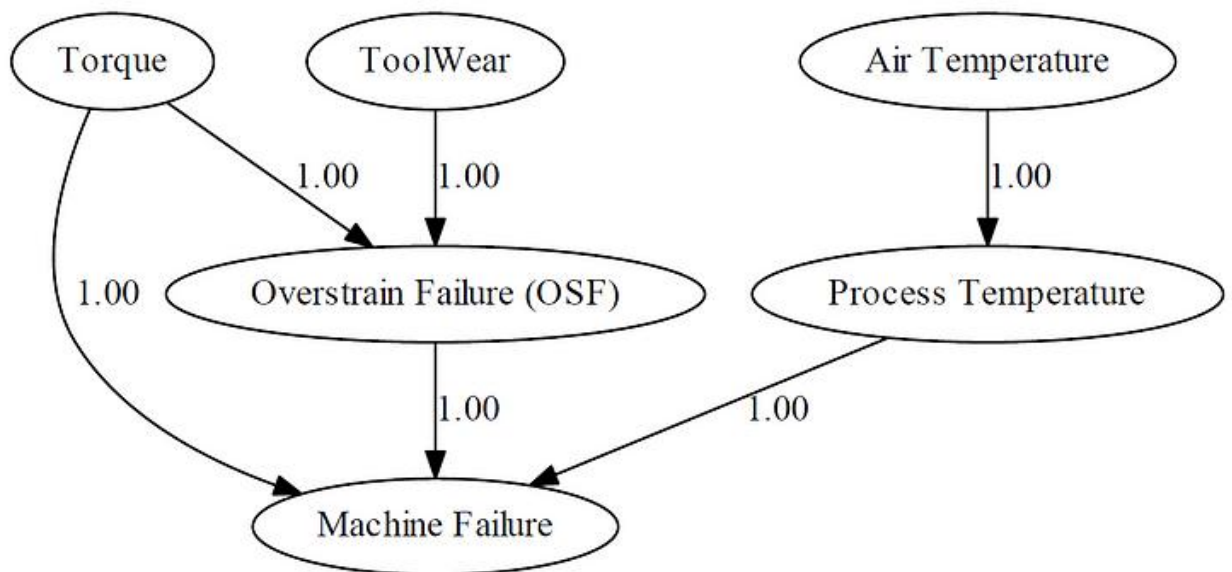


Рисунок 3.1 – DAG системи передбачувального обслуговування

- Крутний момент → Відмова машини;
- Крутний момент → Відмова через перевантаження (OSF);
- Знос інструмента → Відмова через перевантаження (OSF);
- Температура повітря → Температура процесу;

– Відмова через перевантаження (OSF) → Відмова машини.

Побудову відповідного DAG наведено у Лістингу А.13 (Додаток А).

### 3.5 CPT для системи технічного обслуговування

Для кожного з шести вузлів визначаються умовно-ймовірнісні таблиці на основі знань експерта:

Температура повітря:  $P(\text{низька})=0,70$ ;  $P(\text{висока})=0,30$  – без батьківських залежностей;

Знос інструмента:  $P(\text{низький})=0,80$ ;  $P(\text{високий})=0,20$  – без батьківських залежностей;

Крутний момент:  $P(\text{нормальний})=0,90$ ;  $P(\text{високий})=0,10$  – без батьківських залежностей;

Температура процесу: залежить від температури повітря; за низької температури повітря – 70% шансів низької температури процесу;

Відмова через перевантаження (OSF): залежить від крутного моменту та зносу інструмента; за обох факторів у критичному стані ризик OSF становить 90%;

Відмова машини: найскладніший вузол із трьома батьківськими залежностями; найвища ймовірність відмови (80%) при одночасно критичних значеннях усіх трьох чинників.

Повний код визначення та об'єднання CPT із DAG для системи передбачувального обслуговування наведено у Лістингу А.14 (Додаток А).

### 3.6 Генерація синтетичних даних

Побудована модель може бути використана для генерації синтетичних категоріальних даних, що імітують знання експерта. Функція `bn.sampling()` дозволяє згенерувати довільну кількість вибірок. Результатом є повний набір даних з усіма залежними змінними, що може використовуватися для тестування

алгоритмів машинного навчання, бенчмаркінгу або аугментації даних (Лістинг А.15, Додаток А).

### 3.7 Бібліотека `bnlearn`

Бібліотека `bnlearn` є відкритим програмним інструментом (ліцензія MIT) для Python, призначеним для створення, аналізу та візуалізації байєсівських мереж. Вона підтримує дискретні, змішані та неперервні набори даних і охоплює весь спектр основних байєсівських конвеєрів. Бібліотека налічує понад 500 зірок на GitHub та більше 600 000 завантажень.

Ключові функціональні можливості бібліотеки:

- структурне навчання (Structure Learning) – навчання структури мережі з даних або інтеграція знань експерта;
- навчання параметрів (Parameter Learning) – оцінка умовних розподілів ймовірностей на основі спостережених даних;
- інференс (Inference) – обчислення ймовірностей, моделювання інтервенцій та тестування причинних ефектів;
- генерація синтетичних даних (Synthetic Data) – генерація нових наборів даних із навчених байєсівських мереж;
- дискретизація даних (Discretize Data) – перетворення неперервних змінних у дискретні стани;
- оцінка моделі (Model Evaluation) – порівняння мереж за допомогою функцій оцінки, статистичних тестів та показників якості;
- візуалізація (Visualization) – інтерактивне побудування графів, теплові карти ймовірностей та накладення структур.

Переваги `bnlearn` порівняно з іншими реалізаціями байєсівського аналізу:

- містить найбільш затребувані байєсівські конвеєри;
- простий та інтуїтивний у використанні;
- відкритий код із ліцензією MIT;
- наявність документації та навчальних матеріалів.

### 3.8 Обмеження та практичні рекомендації

Незважаючи на широкі можливості, байєсівські мережі та підхід, що базується на знаннях, мають суттєві обмеження, які необхідно враховувати на практиці.

Складність CPT зростає експоненціально зі збільшенням кількості батьківських вузлів. Якщо вузол має  $m$  батьківських булевих змінних, то таблиця ймовірностей містить  $2^m$  записів. Для реальних систем із великою кількістю залежностей це стає значним когнітивним бар'єром для експерта. Рекомендується обмежуватися графами до 10–15 вузлів.

Суб'єктивність ймовірностей є ще одним важливим обмеженням: ймовірності, одержані від опитування експерта, є суб'єктивними та відображають переконання конкретної особи в конкретних умовах. Через це одна й та сама система може бути коректно описана різними (і навіть суперечливими) моделями. Необхідно нормалізувати вербальні формулювання ймовірностей і враховувати контекстну залежність їх тлумачення.

Для підвищення достовірності моделі рекомендується: емпірично перевіряти відношення умовної незалежності між змінними; інтегрувати апріорні знання (заздалегідь визначені DAG або CPT) у навчання; поєднувати структуру, визначену на основі знань, із параметрами, навченими з даних.

Пошук оптимального DAG є обчислювально затратним – виснажливий пошук обмежується вже при  $\sim 15$  вузлах. Для великих наборів даних з сотнями або тисячами змінних доцільно використовувати деревоподібні або обмеження-базовані підходи.

Практичні рекомендації щодо роботи з експертами: проводити декілька коротких інтерв'ю замість одного довгого; формулювати запитання систематично (спочатку граф, потім CPT); бути обережними при обговоренні ймовірностей; розуміти, як експерт виводить свої оцінки, і нормалізувати при необхідності; перевіряти, чи може місце і час змінити результат; проводити перевірки розумності після побудови моделі.

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

В умовах прискороного розвитку інформаційних технологій, професії у сфері ІТ займають ключове місце в економічній структурі та соціальному устрої. Відповідно, набуває актуальності забезпечення адекватних умов праці та охорони здоров'я для спеціалістів цієї галузі. Попри загальноприйнятту думку про порівняно низький ризик в роботі програмістів порівняно з іншими професіями, ця сфера містить специфічні ризики та особливості, що вимагають індивідуального підходу до питань охорони праці.

Аналіз робочого середовища ІТ-спеціалістів показує, що, незважаючи на зовнішню зручність, існують недоліки з точки зору ергономіки, психологічного навантаження та інших важливих аспектів. У цьому контексті важливим є вивчення основних аспектів охорони праці програмістів, аналіз потенційних ризиків та розробка рекомендацій щодо оптимізації умов праці для фахівців у галузі інформаційних технологій.

### **4.1 Питання щодо охорони праці**

Фактори трудового середовища можуть істотно впливати на здоров'я та працездатність розробника програм. Неправильно організоване робоче місце може викликати проблеми із хребтом, шиєю, зап'ястям та іншими частинами тіла. Довгий час роботи за комп'ютером може призвести до тунельного синдрому зап'ястного каналу.

#### **4.1.1 Ергономіка**

У сучасних умовах трудової діяльності значуще місце займає організація робочого місця користувачів персональних комп'ютерів. Законодавство України наголошує на необхідності забезпечення безпечних та комфортних умов праці. Зокрема, відповідно до КЗпП та Закону України "Про охорону праці",

роботодавці мають обов'язки стосовно забезпечення працівників належними умовами.

При організації робочих місць із персональними комп'ютерами важливо забезпечити відстані між боковими частинами столів не менше ніж 1,2 м, а також враховувати, що відстань між задньою частиною одного комп'ютера та екраном іншого має бути 2,5 м. Структура робочого столу повинна бути розроблена відповідно до ергономічних стандартів, дозволяючи ефективно розташовувати необхідне обладнання, таке як дисплей, клавіатура, принтер, а також робочі документи.

Щодо параметрів робочого столу: його висота повинна знаходитися у діапазоні від 680 до 800 мм. Що стосується ширини та глибини, то вони повинні бути такими, щоб працівник міг з легкістю працювати в зоні доступності рук (для цього рекомендовані показники: ширини від 600 до 1400 мм, глибини від 800 до 1000 мм). Також необхідно передбачити комфортний простір для ніг користувача: висота – мінімум 600 мм, ширина – мінімум 500 мм, глибина на рівні колін – 450 мм і на рівні витягнутої ноги – не менше 650 мм.

Стілець на робочому місці повинен мати підйомно-поворотні характеристики, можливість регулювання по висоті, куту нахилу сидушки і спинки, а також відстані від спинки до зовнішнього краю сидіння. Поверхня сидіння має бути рівною, а її зовнішній край має мати округлу форму. Налаштування по кожному параметру повинно бути індивідуальним, інтуїтивним та надійно фіксуватися.

Інтервал регулювання частин стільця: для лінійних розмірів – 15-20 мм, для кутових 2-5°. Зусилля для регулювання не має перевищувати 20 Н. Висота сидіння повинна бути від 400 до 500 мм, а її ширина і глибина – не менше 400 мм. Сидіння має мати можливість нахилу до 15° вперед і до 5° назад. Висота спинки стільця –  $300 \pm 20$  мм, ширина – не менше 380 мм, радіус кривизни горизонтально – 400 мм. Кут нахилу спинки може регулюватися в діапазоні 1-30° від вертикального положення. Відстань від спинки до сидіння – від 260 до 400 мм. Для зменшення напруження в руках рекомендується використовувати

підлокітники довжиною від 250 мм, шириною 50-70 мм, що налаштовуються по висоті від 230 до 260 мм та по відстані між ними від 350 до 500 мм. Матеріал сидіння і спинки має бути напівтвердим, антиковзним, що пропускає повітря і легко миється, а також не збирає статичний заряд.

Робоча зона повинна бути обладнана підніжкою шириною мінімум 300 мм, глибиною не менше 400 мм, з можливістю регулювання по висоті до 150 мм і нахилом до 20°. На підніжці має бути рельєфна поверхня і маленький борт по зовнішньому краю висотою 10 мм. Екран комп'ютера слід розміщувати на оптимальній відстані від користувача, що варіюється в межах 500-700 мм, але не ближче ніж 500 мм, з урахуванням легкості читання тексту і зображень. Екран повинен бути розташований так, щоб забезпечити комфорт при спостереженні, кутом у 30° від вертикалі.

#### **4.1.2 Освітлення**

Ефективне та грамотне виробниче світло підвищує якість зорової діяльності, зменшує втомленість, стимулює зростання продуктивності, сприяє комфортному робочому середовищу, додаючи спокій та позитив працівникам, а також підсилює безпеку роботи, зменшуючи ризик травм. Недостатнє або надто яскраве освітлення може спричинити зорове перевтомлення і головний біль.

Недолік світла може призводити до перевантаження очей, зниження уваги та передчасної втоми. Занадто яскраве світло викликає сліпоту, незадоволення та відчуття дискомфорту в очах. Неправильне розташування світла може утворювати відблиски, контрастні тіні та дезорієнтувати працівника. Ці фактори можуть спричинити нещасний випадок або професійні захворювання, тому важливо правильно розраховувати інтенсивність світла.

Розрізняють три типи світла - природне, штучне та комбіноване.

Природне світло – це варіант освітлення, при якому денне світло проникає через вікна або інші прозорі елементи приміщення. Інтенсивність природного світла може сильно коливатися в залежності від доби, сезону та інших факторів.

Штучне світло використовують у вечірній час або коли природне світло недостатнє. Якщо природне світло доповнюється штучним, таке освітлення називають змішаним.

За своїм призначенням штучне світло може бути робочим, евакуаційним, аварійним чи охоронним. Робоче світло поділяється на загальне та місцеве. При загальному освітленні світильники розташовані рівномірно по приміщенню. У системі комбінованого освітлення до загального додається додаткове місцеве світло.

Згідно норм "ДБН В.2.5-28:2018 Природне і штучне освітлення", в комп'ютерних залах слід застосовувати комбінований тип освітлення. Для виконання робіт високої зорової точності (об'єкт розрізнення 0,3 ... 0,5 мм) інтенсивність природного світла має бути не менше 1,5%. Для робіт середньої точності (об'єкт розрізнення 0,5 ... 1,0 мм) - не менше 1,0%. Як джерела штучного світла часто використовують лампи типу ЛБ або ДРЛ, об'єднані в світильниках над робочими столами.

Вимоги до освітленості для робочих місць з комп'ютерами такі: при високій точності зорової роботи – 300 лк (загальне) та 750 лк (комбіноване); при середній точності – 200 та 300 лк відповідно.

Також важливо, щоб усе поле зору було якісно освітлене. Світлова інтенсивність приміщення та яскравість екрану комп'ютера повинні бути близькими, адже надлишкове світло може збільшувати втомленість очей.

### **4.1.3 Параметри мікроклімату**

Для створення комфортних умов праці в приміщеннях з комп'ютерною технікою важливо дотримуватися певних стандартів мікроклімату. В таблиці 4.1 наведені параметри мікроклімату для таких приміщень

Таблиця 4.1 – Параметри мікроклімату для приміщень, де встановлені комп'ютери

Параметри мікроклімату	Літній період	Зимовий період
Температура повітря, °С	20 – 24	22 – 24
Відносна вологість, %	40 – 60	40 – 60
Швидкість руху повітря, м/с	0,1 – 0,2	0,1 – 0,2
Рівень шуму, дБ	до 50	До 50

#### 4.1.4 Емоційна психогігієна

У сфері ІТ охорона праці включає не лише фізичне, але й психічне здоров'я працівників. Фактори, що впливають на психіку працівників у галузі інформаційних технологій, є різноманітними та мають велике значення у створенні здорового робочого середовища.

Таблиця 4.2 – Допустимі значення параметрів неіонізуючого електромагнітного випромінювання

Найменування параметра	Допустимі значення
Напруженість електричної складової електромагнітного поля на відстані 50см від поверхні монітора	10 В / м
Напруженість магнітної складової електромагнітного поля на відстані 50см від поверхні монітора	0,3 А / м
Напруженість електростатичного поля	20кВ / м

До них належать:

– тривалість та інтенсивність робочого дня: довгі години роботи без відпочинку можуть призвести до перевтоми та стресу, впливаючи на психічне здоров'я;

- терміни виконання завдань і робочий тиск: нереалістичні терміни виконання завдань або надмірний робочий тиск можуть викликати стрес та тривогу;
- міжособистісні взаємодії: конфлікти в команді або нездорова робоча атмосфера можуть спричиняти емоційне вигорання;
- робота в умовах ізоляції або віддалена робота: відсутність соціальної взаємодії або підтримки може вплинути на почуття ізоляції та самотності;
- work-life баланс: нездатність знайти баланс між роботою та особистим життям може призвести до стресу та емоційного вигорання;
- значне розумове навантаження: постійна потреба в освоєнні нових технологій та адаптації до змін може бути стресовою;
- відсутність автономії або контролю над роботою: обмежений контроль над робочими процесами та відсутність автономії можуть викликати почуття безпорадності та фрустрації.

Для ефективної профілактики психоемоційних проблем, пов'язаних з роботою в ІТ-галузі, необхідно розробити комплексний підхід, який враховує різноманітність факторів, що впливають на психічне здоров'я працівників. Значущість цього підходу полягає в його спрямованості на зменшення стресорів у робочому середовищі та підвищення ресурсів для психічного відновлення.

Першочергово, акцентується увага на регулюванні тривалості робочого дня та інтенсивності роботи. Це включає розробку ефективних графіків роботи, що передбачають достатні перерви для відновлення, та уникнення перевантаження завданнями. Крім того, розглядається важливість балансу між роботою та особистим життям, який може бути підтриманий через гнучкі робочі години та можливість дистанційної роботи.

Ще одним важливим аспектом є оптимізація робочого середовища з точки зору ергономіки. Покращення умов роботи на робочому місці, включаючи забезпечення якісного обладнання та комфортних умов, сприятиме зниженню фізичного та психологічного дискомфорту.

Ключову роль відіграє також розвиток корпоративної культури, яка підтримує психологічне благополуччя. Це передбачає створення відкритого, підтримуючого співробітництва та комунікації середовища, заохочення взаємодопомоги між колегами, та реалізацію програм з управління стресом.

Для забезпечення довгострокового психічного благополуччя, розглядається імплементація регулярних тренінгів та навчальних програм, спрямованих на розвиток навичок управління стресом та підвищення особистісної стійкості. Також важливим є застосування систематичного моніторингу психічного стану працівників з використанням психометричних інструментів для раннього виявлення потенційних проблем.

Розуміння та належне врахування цих факторів є ключовим для забезпечення психічного благополуччя працівників у галузі ІТ, що, у свою чергу, позитивно впливає на їх продуктивність та загальне задоволення роботою.

#### **4.2 Питання щодо безпеки в надзвичайних ситуаціях**

В умовах швидкого розвитку інформаційних технологій та зростання залежності бізнес-процесів від ІТ-систем, питання безпеки в сфері ІТ набуває особливої актуальності. Розглядаючи безпеку праці в ІТ, особливу увагу необхідно приділити розробці та імплементації процедур реагування на надзвичайні ситуації, що включають технічні збої, кібератаки, витік конфіденційної інформації, фізичні загрози обладнанню та перебої в електропостачанні, а також інші аварійні стани, здатні порушити звичний хід робочих процесів.

Відповідальність за реалізацію ефективної системи безпеки лежить на ІТ-спеціалістах, керівництві компаній та інших зацікавлених сторонах. Необхідно забезпечити встановлення чітких правил та інструкцій, які регламентують дії персоналу у випадку виникнення надзвичайних ситуацій. Це включає визначення швидких комунікаційних каналів, розробку планів відновлення

роботи систем, а також проведення регулярних тренінгів та інструктажів з охорони праці для всіх працівників.

Плани евакуації та реагування мають бути адаптовані до можливих ІТ-ризиків. Вони повинні включати дії для захисту життя та здоров'я працівників, а також процедури збереження даних та обладнання. Плани мають бути детальними та зрозумілими для кожного члена команди, з чіткими інструкціями щодо дій в кризових ситуаціях.

Регулярні тренінги з охорони праці, які охоплюють надзвичайні ситуації в ІТ, є необхідними для підготовки персоналу до ефективного реагування на інциденти. Навчальні програми повинні включати імітації надзвичайних ситуацій, навчання з використанням спеціалізованого обладнання та вивчення найкращих практик у сфері кібербезпеки. Систематичний аналіз ризиків є важливим для ідентифікації потенційних вразливостей системи та розробки відповідних заходів запобігання. Встановлення систем моніторингу, які можуть виявити ознаки надзвичайних ситуацій на ранніх стадіях, допомагає зменшити потенційні збитки та забезпечити оперативне реагування.

Після кожної надзвичайної ситуації необхідно проводити детальний аналіз її причин, ефективності дій персоналу та наявних процедур реагування. На основі цього аналізу слід вносити корективи у плани надзвичайних дій, поліпшувати процедури безпеки та організовувати додаткове навчання. Ефективне управління надзвичайними ситуаціями в ІТ-сфері вимагає всебічного підходу, який поєднує в собі як технічні, так і організаційні аспекти. Відповідальність за це лежить на всіх рівнях організації, починаючи з ІТ-фахівців і закінчуючи вищим керівництвом. Завдяки встановленню та дотриманню відповідних процедур можна мінімізувати ризики та забезпечити безперебійну роботу в умовах, що постійно змінюються.

## ВИСНОВКИ

У цій роботі систематично викладено ключові концепції та практичні методи роботи з байєсівськими мережами та причинно-наслідковим моделюванням засобами Python. На основі матеріалу, що охоплює теоретичне підґрунтя, практичні приклади та критичний аналіз обмежень, можна сформулювати такі основні висновки.

По-перше, знання людини є повноцінним видом даних. Будь-яке експертне судження, причинне твердження або евристичне правило може бути формалізоване у вигляді змінних, зв'язків та ймовірностей, а отже – використане в обчислювальних моделях. Байєсівські мережі забезпечують для цього природний і математично обґрунтований механізм.

По-друге, ефективна побудова знання-керованої моделі вимагає двох компонентів: спрямованого ациклічного графа (DAG), що кодує структурні припущення про систему, та умовно-ймовірнісних таблиць (CPT), що кількісно описують силу залежностей між вузлами. Разом ці компоненти формують повну ймовірнісну специфікацію системи.

По-третє, байєсівське структурне навчання дозволяє автоматично відновлювати причинно-наслідкову структуру з даних. При цьому важливо розрізняти кореляцію, асоціацію та причинність: лише визначення напрямку ребер у DAG та аналіз умовної незалежності між змінними дозволяють робити обґрунтовані причинні висновки.

По-четверте, поєднання знань експерта зі статистичним навчанням з даних є найбільш потужним підходом. Структура, визначена на основі знань, може бути уточнена за допомогою навчання параметрів, а синтетичні дані, згенеровані з побудованої моделі, є ефективним інструментом для тестування та аугментації.

По-п'яте, практичне застосування байєсівських мереж пов'язане з рядом принципових обмежень: суб'єктивністю експертних оцінок, експоненціальним зростанням розмірності CPT зі збільшенням кількості батьківських вузлів та обчислювальною складністю пошуку оптимального DAG. Подолання цих

обмежень потребує ретельної підготовки інтерв'ю з експертами, обґрунтованого вибору алгоритмів пошуку та своєчасних перевірок розумності моделі.

Бібліотека `bnlearn` для Python є зручним і потужним інструментом, що охоплює повний цикл роботи з байєсівськими мережами: від побудови структури та визначення параметрів до здійснення інференсу та генерації синтетичних даних. Вона рекомендується як основний інструмент для дослідників і практиків у даній галузі.

Окремо слід наголосити, що наскрізним лейтмотивом усієї роботи є ідея людино-машинної взаємодії як продуктивного партнерства, а не протиставлення людини й алгоритму. Проведене дослідження демонструє, що байєсівські мережі є чи не найприроднішим середовищем для такої співпраці: вони здатні однаково сприймати «м'які» знання людини, виражені у формі переконань, евристик та причинних суджень, і «жорсткі» статистичні закономірності, відновлені з даних. У цій моделі взаємодії ролі людини та машини є взаємодоповнювальними – експерт привносить контекст, розуміння причинності та здатність міркувати в умовах неповної інформації, тоді як обчислювальний апарат забезпечує строгість, нормалізацію ймовірностей та здатність опрацювати велику кількість залежностей. Спрямований ациклічний граф і умовно-ймовірнісні таблиці виконують при цьому роль спільного інтерфейсу, достатньо формального для машинного опрацювання й водночас достатньо інтуїтивного, щоб експерт міг його осмислити та скоригувати. Саме така синергія, у якій збережено інтерпретованість і причинне міркування людини та водночас додано обчислювальну строгість машини, робить байєсівське моделювання особливо цінним у галузях із високою ціною помилки та обмеженим обсягом даних, і відкриває перспективу для ітеративного діалогу, у якому модель поступово уточнюється у тісній взаємодії експерта й системи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wikipedia. Knowledge. URL: <https://en.wikipedia.org/wiki/Knowledge>
2. Pearl, Judea (2000). Causality: Models, Reasoning, and Inference. Cambridge University Press. ISBN 978-0-521-77362-1.
3. Taskesen E. The Starters Guide to Causal Structure Learning with Bayesian Methods in Python. Data Science Collective (DSC), September 2025.
4. Willems S., et al. Variability in the interpretation of probability phrases used in Dutch news articles – a risk for miscommunication. JCOM, 24 March 2020.
5. Jeffrey R. Subjective Probability: The Real Thing. Cambridge University Press, 2004.
6. Tversky A., Kahneman D. Judgment under Uncertainty: Heuristics and Biases. Science, 1974.
7. Tversky A., Kahneman D. Judgment under uncertainty: Heuristics and biases // Judgment under Uncertainty. Cambridge University Press, 1982. P. 3–20.
8. Das B. Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem. arXiv:cs/0411034.
9. Taskesen E. Synthetic Data: The Essentials Of Data Generation Using Bayesian Sampling. Data Science Collective (DSC), May 2025.
10. Taskesen E. Why Prediction Isn't Enough: Using Bayesian Models to Change the Outcome. Data Science Collective (DSC), June 2025.
11. McLeod S.A. Correlation: definitions, examples & interpretation. Simply Psychology, 14 January 2018.
12. Dablander F. An Introduction to Causal Inference. University of Amsterdam. URL: <https://psyarxiv.com/b3fkw>
13. Davis B. When Correlation is Better than Causation. Medium, 2021.
14. Gingrich P. Measures of association. P. 766–795. URL: <http://uregina.ca/~gingrich/ch11a.pdf>
15. Taskesen E. Uncover Hidden Patterns In Your Tabular Datasets. Data Science Collective (DSC), June 2025.

16. Holländer B. Introduction to Probabilistic Graphical Models. Medium, 2020.
17. Padmanaban H. Comparative Analysis of Naive Bayes and Tree Augmented Naive Bayes Models. San Jose State University, 2014.
18. Huszar F. ML beyond Curve Fitting: An Intro to Causal Inference and do-Calculus. URL: <https://www.inference.vc/untitled/>
19. AI4I 2020 Predictive Maintenance Dataset. (2020). UCI Machine Learning Repository. DOI: 10.24432/C5HS5C. Licensed under CC BY 4.0.
20. Perrier E., et al. Finding Optimal Bayesian Network Given a Super-Structure. Journal of Machine Learning Research 9, 2008. P. 2251–2286.
21. Голінько В. І. Охорона праці в галузі інформаційних технологій: навч. посіб. / В. І. Голінько, М. Ю. Іконніков, Я. Я. Лебедєв; М-во освіти і науки України, Держ. вищий навч. закл. "Нац. гірн. ун-т". - Дніпропетровськ: НГУ, 2015. - 246 с.
22. Бабак, В. П., Марченко, Б. Г., & Фриз, М. Е. (2004). Теорія ймовірностей, випадкові процеси та математична статистика.
23. Капаціла М. І. Використання формули Баєса для фільтрування спаму / Капаціла М. І. // Збірник тез Міжнародної студентської науково-технічної конференції „Природничі та гуманітарні науки. Актуальні питання“, 26-27 квітня 2018 року. – Т. : ТНТУ, 2018. – Том 1. – С. 66. – (Математика).
24. Гандзюк М.П. Основи охорони праці: Підручник. 4-е вид./Гандзюк М.П., Желібо Є.П., Халімовський М.О. - Київ: Каревела, 2008. – 384с.
25. Техноекологія та цивільна безпека. Частина «Цивільна безпека»: Навчальний посібник; укл.: Стручок В. С. Тернопіль: ФОП Паляниця В.А., 2022. 150 с.
26. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

27. Умови праці працівників, які використовують у роботі персональні комп'ютери. Zolochiv.Net. URL: <https://zolochiv.net/umovy-pratsi-pratsivnykiv-iaki-vykorystovuiut-u-roboti-personal-ni-komp-iutery/> (дата звернення: 25.10.2024).

# ДОДАТКИ

## Програмний код

**Лістинг А.1. Перевірка асоціації між відмовою машини та PWF (гіпергеометричний тест)**

```
# Встановлення бібліотеки
pip install bnlearn

import bnlearn as bn
import pandas as pd
from scipy.stats import hypergeom

# Завантаження набору даних передбачувального обслуговування
df = bn.import_example(data='predictive_maintenance')

# Перевірка асоціації між machine failure і PWF
# H0: між machine failure і power failure (PWF) асоціація відсутня
M = df.shape[0] # розмір популяції
n = sum(df['Machine failure'] == 1) # кількість відмов машини
N = sum(df['PWF'] == 1) # кількість подій PWF
x = sum((df['PWF'] == 1) & (df['Machine failure'] == 1)) # перетин

# Обчислення P(X >= x)
P = hypergeom.sf(x - 1, M, n, N)
print(P) # 1.66e-146 → H0 відхиляється
```

*Лістинг А.1 – розділ 1.1*

**Лістинг А.2. Побудова DAG для системи зрошення (Sprinkler)**

```
import bnlearn as bn

# Визначення причинних залежностей
edges = [('Cloudy', 'Sprinkler'),
         ('Cloudy', 'Rain'),
         ('Sprinkler', 'Wet_Grass'),
         ('Rain', 'Wet_Grass')]
```

```

# Побудова DAG
DAG = bn.make_DAG(edges)

# Статичний графік
bn.plot(DAG)

# Інтерактивний графік (HTML)
bn.plot(DAG, interactive=True)

# Матриця суміжності
print(DAG["adjmat"])

```

*Лістинг А.2 – розділ 2.2*

### **Лістинг А.3. СРТ вузла "Хмарно"**

```

from pgmpy.factors.discrete import TabularCPD

cpt_cloudy = TabularCPD(
    variable='Cloudy',
    variable_card=2,
    values=[[0.3], # P(Cloudy=0) - ясно
            [0.7]] # P(Cloudy=1) - хмарно
)

print(cpt_cloudy)

```

*Лістинг А.3 – розділ 2.3*

### **Лістинг А.4. СРТ вузла "Дощ"**

```

cpt_rain = TabularCPD(
    variable='Rain',
    variable_card=2,
    values=[[0.8, 0.2], # P(Rain=0 | Cloudy=0/1)
            [0.2, 0.8]], # P(Rain=1 | Cloudy=0/1)
    evidence=['Cloudy'],
    evidence_card=[2]
)

print(cpt_rain)

```

*Лістинг А.4 – розділ 2.3*

### Лістинг А.5. CPT вузла "Спринклер"

```
cpt_sprinkler = TabularCPD(  
    variable='Sprinkler',  
    variable_card=2,  
    values=[[0.5, 0.9], # P(Sprinkler=0 | Cloudy=0/1)  
            [0.5, 0.1]], # P(Sprinkler=1 | Cloudy=0/1)  
    evidence=['Cloudy'],  
    evidence_card=[2]  
)  
print(cpt_sprinkler)
```

*Лістинг А.5 – розділ 2.3*

### Лістинг А.6. CPT вузла "Мокра трава"

```
cpt_wet_grass = TabularCPD(  
    variable='Wet_Grass',  
    variable_card=2,  
    values=[[0.01, 0.1, 0.1, 1.0], # P(WG=0 | Rain, Sprinkler)  
            [0.99, 0.9, 0.9, 0.0]], # P(WG=1 | Rain, Sprinkler)  
    evidence=['Rain', 'Sprinkler'],  
    evidence_card=[2, 2]  
)  
print(cpt_wet_grass)
```

*Лістинг А.6 – розділ 2.3*

### Лістинг А.7. Об'єднання DAG із CPT та верифікація моделі спринклера

```
import bnlearn as bn  
  
# Підключення CPT до DAG  
model = bn.make_DAG(DAG, CPD=[cpt_cloudy, cpt_rain, cpt_sprinkler,  
cpt_wet_grass])  
  
# Виведення CPT  
bn.print_CPD(model)
```

*Лістинг А.7 – розділ 2.3*

### Лістинг А.8. Інференс на моделі спринклера (знання-керована модель)

```
# Запит 1: P(Wet_Grass | Sprinkler=0)
```

```

q1 = bn.inference.fit(model, variables=['Wet_Grass'],
evidence={'Sprinkler': 0})
# Wet_Grass=1: 61.6%

# Запит 2: P(Rain | Sprinkler=0, Cloudy=1)
q2 = bn.inference.fit(model, variables=['Rain'], evidence={'Sprinkler': 0,
'Cloudy': 1})
# Rain=1: 80.0%

# Запит 3: P(Wet_Grass, Rain | Sprinkler=1)
q3 = bn.inference.fit(model, variables=['Wet_Grass', 'Rain'],
evidence={'Sprinkler': 1})
# Wet_Grass=1 & Rain=1: 38.7%

```

*Лістинг А.8 – розділ 2.4*

### **Лістинг А.9. Структурне навчання для набору даних "Спринклер"**

```

import bnlearn as bn

# Завантаження набору даних
df = bn.import_example('sprinkler')

# Навчання структури (метод HillClimbSearch, оцінка BIC)
DAG = bn.structure_learning.fit(df)

# Варіанти методів та оцінок
model_hc_bic = bn.structure_learning.fit(df, methodtype='hc',
scoretype='bic')
model_hc_k2 = bn.structure_learning.fit(df, methodtype='hc',
scoretype='k2')
model_ex_bic = bn.structure_learning.fit(df, methodtype='ex',
scoretype='bic')
model_cs_bic = bn.structure_learning.fit(df, methodtype='cs',
scoretype='bic')
model_cl = bn.structure_learning.fit(df, methodtype='cl',
root_node='Wet_Grass')

# Матриця суміжності
print(DAG['adjmat'])

```

```
# Побудова графу
G = bn.plot(DAG)
```

*Лістинг А.9 – розділ 3.1*

### **Лістинг А.10. Навчання параметрів методом MLE**

```
import bnlearn as bn

df = bn.import_example('sprinkler')

edges = [('Cloudy', 'Sprinkler'),
         ('Cloudy', 'Rain'),
         ('Sprinkler', 'Wet_Grass'),
         ('Rain', 'Wet_Grass')]

DAG = bn.make_DAG(edges)

# MLE
model = bn.parameter_learning.fit(DAG, df, methodtype='ml')
bn.print_CPD(model)

# Байєсівська оцінка
model_bayes = bn.parameter_learning.fit(DAG, df, methodtype='bayes')
bn.print_CPD(model_bayes)
```

*Лістинг А.10 – розділ 3.2*

### **Лістинг А.11. Ручне обчислення MLE для вузлів "Хмарно" та "Дош"**

```
# P(Cloudy=0)
sum(df['Cloudy']==0) / df.shape[0] # 0.488

# P(Cloudy=1)
sum(df['Cloudy']==1) / df.shape[0] # 0.512

# P(Rain=0 | Cloudy=0)
sum((df['Cloudy']==0) & (df['Rain']==0)) / sum(df['Cloudy']==0) # 0.807
```

```

# P(Rain=1 | Cloudy=0)
sum((df['Cloudy']==0) & (df['Rain']==1)) / sum(df['Cloudy']==0) # 0.193

# P(Rain=0 | Cloudy=1)
sum((df['Cloudy']==1) & (df['Rain']==0)) / sum(df['Cloudy']==1) # 0.178

# P(Rain=1 | Cloudy=1)
sum((df['Cloudy']==1) & (df['Rain']==1)) / sum(df['Cloudy']==1) # 0.822

```

*Лістинг A.11 – розділ 3.2*

### **Лістинг A.12. Інференс на моделі, навченій з даних (Байєсівська оцінка)**

```

import bnlearn as bn

df = bn.import_example('sprinkler')
edges = [('Cloudy', 'Sprinkler'), ('Cloudy', 'Rain'),
         ('Sprinkler', 'Wet_Grass'), ('Rain', 'Wet_Grass')]
DAG = bn.make_DAG(edges)
model = bn.parameter_learning.fit(DAG, df, methodtype='bayes')

# P(Wet_Grass | Sprinkler=0)
q1 = bn.inference.fit(model, variables=['Wet_Grass'],
evidence={'Sprinkler': 0})
# Wet_Grass=1: 51.3%

# P(Rain | Sprinkler=0, Cloudy=1)
q2 = bn.inference.fit(model, variables=['Rain'], evidence={'Sprinkler': 0,
'Cloudy': 1})
# Rain=1: 66.3%

# P(Wet_Grass, Rain | Sprinkler=1)
q3 = bn.inference.fit(model, variables=['Wet_Grass', 'Rain'],
evidence={'Sprinkler': 1})
# Rain=1 & Wet_Grass=1: 28.8%

```

*Лістинг A.12 – розділ 3.3*

### **Лістинг A.13. DAG для системи передбачувального обслуговування**

```

import bnlearn as bn

```

```

edges = [
    ('Process Temperature', 'Machine Failure'),
    ('Torque', 'Machine Failure'),
    ('Torque', 'Overstrain Failure (OSF)'),
    ('Tool Wear', 'Overstrain Failure (OSF)'),
    ('Air Temperature', 'Process Temperature'),
    ('Overstrain Failure (OSF)', 'Machine Failure'),
]

DAG = bn.make_DAG(edges)
bn.plot(DAG)
dotgraph = bn.plot_graphviz(DAG, edge_labels='pvalue')
dotgraph.view(filename='predictive_maintenance_dag.pdf')

```

*Лістинг A.13 – розділ 3.4*

#### **Лістинг A.14. CPT та об'єднання з DAG для системи передбачувального обслуговування**

```

from pgmpy.factors.discrete import TabularCPD
import bnlearn as bn

# Температура повітря (немає батьків)
cpt_air_temp = TabularCPD(
    variable='Air Temperature', variable_card=2,
    values=[[0.7], [0.3]]) # низька / висока

# Знос інструмента (немає батьків)
cpt_toolwear = TabularCPD(
    variable='Tool Wear', variable_card=2,
    values=[[0.8], [0.2]])

# Крутний момент (немає батьків)
cpt_torque = TabularCPD(
    variable='Torque', variable_card=2,
    values=[[0.9], [0.1]])

# Температура процесу (залежить від температури повітря)
cpt_process_temp = TabularCPD(

```

```

    variable='Process Temperature', variable_card=2,
    values=[[0.7, 0.2],
            [0.3, 0.8]],
    evidence=['Air Temperature'],
    evidence_card=[2])

# Відмова через перевантаження (OSF)
cpt_osf = TabularCPD(
    variable='Overstrain Failure (OSF)', variable_card=2,
    values=[[0.9, 0.5, 0.7, 0.1],
            [0.1, 0.5, 0.3, 0.9]],
    evidence=['Torque', 'Tool Wear'],
    evidence_card=[2, 2])

# Відмова машини
cpt_machine_fail = TabularCPD(
    variable='Machine Failure', variable_card=2,
    values=[[0.9, 0.7, 0.6, 0.3, 0.8, 0.5, 0.4, 0.2],
            [0.1, 0.3, 0.4, 0.7, 0.2, 0.5, 0.6, 0.8]],
    evidence=['Process Temperature', 'Torque', 'Overstrain Failure (OSF)'],
    evidence_card=[2, 2, 2])

# Об'єднання DAG із CPT
model = bn.make_DAG(DAG, CPD=[
    cpt_process_temp, cpt_machine_fail,
    cpt_torque, cpt_osf, cpt_toolwear, cpt_air_temp
])

bn.print_CPD(model)

```

*Лістинг A.14 – розділ 3.5*

### **Лістинг A.15. Генерація синтетичних даних із моделі**

```

import bnlearn as bn

# Генерація 1000 синтетичних спостережень
X = bn.sampling(model, n=1000, methodtype='bayes')
print(X)

```

*Лістинг A.15 – розділ 3.6*

Тези доповіді

Міністерство освіти і науки України  
Тернопільський національний технічний університет  
імені Івана Пулюя  
Маріборський університет (Словенія)  
Технічний університет в Кошице (Словаччина)  
Каунаський технологічний університет (Литва)  
Львівський національний університет  
імені Івана Франка  
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)  
Луцький національний технічний університет  
Чернівецький національний університет  
імені Юрія Федьковича  
Вроцлавський економічний університет (Польща)  
Університет технологій та економіки  
імені Хелени Ходковської (Польща)  
Донбаська державна машинобудівна академія



*Студентське наукове  
товариство*



**ІХ МІЖНАРОДНА**

**студентська науково - технічна конференція**

**"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ  
НАУКИ. АКТУАЛЬНІ ПИТАННЯ"**

24-25 квітня 2026 р.

*(збірник тез конференції)*

*Тернопіль 2026*

Шмирко Р. <b>ЦИФРОВА МОДУЛЯЦІЯ У СУЧАСНИХ СИСТЕМАХ РАДІОЗВ'ЯЗКУ</b>	144
Шупа Д. <b>БІОМЕДИЧНА ІНЖЕНЕРІЯ:СУТНІСТЬ, ЗАСТОСУВАННЯ ТА ПЕРСПЕКТИВИ РОЗВИТКУ</b>	146
Яріш Б. <b>КВАНТОВІ КОМП'ЮТЕРИ</b>	148
Яцків О. <b>НАСКІЛЬКИ ВАЖЛИВА ПРОФЕСІЯ РАДІОТЕХНІКА В НАШ ЧАС</b>	150
Андрухов І. <b>РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО ГОЛОСОВОГО АСИСТЕНТА НА БАЗІ ASTERISK ТА GEMINI LIVE API ДЛЯ ПРИЙМАЛЬНОЇ КОМІСІЇ ЗВО</b>	151
Байдецька В. <b>ІМІТАЦІЯ СВІДОМОСТІ У ВЕЛИКИХ МОВНИХ МОДЕЛЯХ(LLM) ТА ПРИЧИНИ ЇЇ СПРИЙНЯТТЯ ЛЮДИНОЮ</b>	152
Бармак Р., Дегодюк І. <b>МЕТОДОЛОГІЯ СПЕЦИФІКАЦІЙНО-ОРІЄНТОВАНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА ДОПОМОГОЮ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ</b>	154
Берестень М. <b>РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СЕРВІСУ УПРАВЛІННЯ ФІНАНСАМИ ПРИВАТНИХ ОСІБ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ FLASK, REACT ТА БАНКІВСЬКИХ API</b>	156
Бица Р. <b>РОЗРОБКА WEB-ЗАСТОСУНКУ УПРАВЛІННЯ АВТОСАЛОНОМ</b>	157
Білінський М. <b>РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ЕНЕРГОСПОЖИВАННЯ КОМЕРЦІЙНИХ БУДІВЕЛЬ НА БАЗІ ТРИРІВНЕВОЇ АРХІТЕКТУРИ З</b>	158
Боб О., Мага С., Лотоцький Д., Боднарчук І. <b>ДО ПИТАННЯ ЯКОСТІ ДАНИХ В НАУКОВИХ ДОСЛІДЖЕННЯХ ТНТУ</b>	159
Боднар Д. <b>СИНЕРГІЯ МЕТОДІВ ОБРОБКИ ПРИРОДНОЇ МОВИ ТА ПОВЕДІНКОВОЇ АНАЛІТИКИ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ МОНІТОРИНГУ</b>	161

УДК 004.6:005.6

Боб О., Мага С., Лотоцький Д., Боднарчук І.

*Тернопільський національний технічний університет імені Івана Пулюя*

## **ДО ПИТАННЯ ЯКОСТІ ДАНИХ В НАУКОВИХ ДОСЛІДЖЕННЯХ ТНТУ**

O.O. Bob, S.Yu. Maha, D.V. Lototskyi, I.O. Bodnarchuk

*Ternopil Ivan Puluj National Technical University*

## **ON THE ISSUE OF DATA QUALITY IN SCIENTIFIC RESEARCH AT TNTU**

Задачею поточного дослідження було систематизувати в загальних рисах наукові публікації з афіліацією Тернопільського національного технічного університету імені Івана Пулюя на тему визначення, забезпечення та покращення якості даних.

Найбільша увага прикута саме до того, як забезпечити якість даних для «мозку» сучасних технологій – систем машинного навчання та штучного інтелекту. У ранніх, але важливих роботах [7] та [10] автори піднімають важливу проблему управління якістю та ефективності ML-систем, без чого загальна якість інформаційних систем суттєво знижується. Дослідження [7] крім теоретичних викладок пропонує також власний метод оцінювання якості даних на основі міжнародного стандарту ISO/IEC 25012, та описує власне програмне рішення для задач контролю якості даних. Робота [8] детально і покроково описує весь процес забезпечення якості для машинного навчання: від профілювання та очищення до стандартизації й усунення дублікатів.

Багато уваги приділяється також процесам підготовки даних. У публікації [6] підготовку та попередню обробку даних подано, як один з найважливіших кроків для підвищення їхньої якості перед подальшою обробкою прогностичними моделями. І це логічно, оскільки якість прогнозу буде високою, коли базові дані матимуть належну якість. У дослідженні [3] автори продовжують цю тему, аналізуючи, як саме параметри моделей впливають на якість класифікації, та розглядаючи методи очищення, нормалізації та масштабування даних.

Окремий частини знайдених робіт присвячена структурній надійності даних та їхній перевірці. На сьогодні ці характеристики даних мають критичну важливість. Наприклад, робота [5] містить огляд можливостей забезпечення цілісності даних в технологіях блокчейн у розподілених системах зберігання. Дуже практичний підхід продемонстровано в [1], де йдеться про використання Django ORM для управління базою даних, що буквально гарантує надійну цілісність даних при автоматизованому зборі наукових публікацій. Також публікація [4] описує реальну проблему: при розробці електронного кабінету абітурієнта ТНТУ виявилось, що однією з основних і найбільших проблем була саме складність отримання та верифікації персональних даних.

Окремої уваги заслуговують менш очевидні, проте не менш значущі напрями досліджень. Зокрема, у роботі [2] розглядається проблема впорядкування метаданих наукових документів: обговорюються підходи до підвищення їхньої повноти та точності шляхом автоматизованого збагачення з відкритих джерел. Дослідження [9], попри зосередженість на тестуванні програмного забезпечення, також торкається аналізу отриманих даних і систем відстеження дефектів – аспектів, що є невід'ємною складовою загального циклу забезпечення якості даних.

Таким чином, можна констатувати, що ТНТУ не лише декларує увагу до проблематики якості даних, а й володіє ґрунтовною дослідницькою базою у цій сфері. Вона охоплює як теоретико-методологічний рівень – зокрема, застосування моделі ISO/IEC 25012 – так і широкий спектр прикладних напрямів: від методів машинного навчання та технологій блокчейну до верифікації даних вступників.

### **Література**

1. Юрчишин Д. І. Розробка інформаційної системи надання наукових сервісів з використанням бібліотеки Scrapy та ORM для взаємодії з базою даних : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 121 - інженерія програмного забезпечення / наук. кер. М. Р. Петрик. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2025. 59 с.
2. Сучков С. С. Система автоматичного формування блоку метаданих наукових метаданих наукових документів з використанням відкритих баз даних : робота на здобуття кваліфікаційного ступеня магістра : спец. 121 - інженерія програмного забезпечення / наук. кер. І. В. Бойко. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2024. 77 с.
3. Микитишин А. Г. Застосування методів машинного навчання для класифікації даних в комп'ютеризованих системах керування / Андрій Григорович Микитишин, І. С. Дідич, Р. І. Яцишин // Тези XIII МНПК „Актуальні задачі сучасних технологій“, 11-12 грудня 2024 року. — Т. : ФОП Паляниця В. А., 2024. — С. 14–16. — (Нові матеріали, міцність і довговічність елементів конструкцій).
4. Карнаухов, О. К. (2024). Дослідження розробки електронного кабінету абітурієнта ТНТУ ім. І. Пулюя. Тези доповідей V міжнародної науково-практичної конференції учених та студентів "Цифрова економіка як фактор інновацій та сталого розвитку суспільства", 46-47.
5. Гладій В. В. Технології створення розподілених комп'ютерних систем зберігання даних на основі блокчейн : кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю "123 — комп'ютерна інженерія" / В. В. Гладій. — Тернопіль: ТНТУ, 2023. — 82 с.
6. Кучеренко О. А. Особливості передоброби даних для методів прогнозування / О. А. Кучеренко, О. О. Кучеренко // ІМСТТ, 13-14 грудня 2023 року. — Т. : ТНТУ, 2023. — С. 72. — (Інформаційні системи та технології, кібербезпека).
7. Журихін Ю. О. Методи забезпечення якості даних при проектуванні систем машинного навчання: автореферат дипломної роботи магістра за спеціальністю 123 «Комп'ютерна інженерія»/ Ю. О. Журихін – Тернопільський національний технічний університет імені Івана Пулюя – Тернопіль, ТНТУ, 2018. – 8 с.
8. Яцишин В. Процеси забезпечення якості даних при проектуванні систем машинного навчання / В. Яцишин, Ю. Журихін // Матеріали V науково-технічної конференції „Інформаційні моделі, системи та технології“, 1-2 лютого 2018 року. — Т. : ТНТУ, 2018. — С. 68. — (Секція 3. Комп'ютерні системи та мережі).
9. Чорновус, Р. М. Визначення якості тестування програмного забезпечення та аналіз отриманих даних. Матеріали конференції. Тернопіль: ТНТУ, 2017. URL: <http://elartu.tntu.edu.ua/handle/123456789/18917>.
10. Яцишин В. В. Оцінювання якості даних для систем машинного навчання / В. В. Яцишин, Ю. О. Журихін // Збірник тез доповідей VI Міжнародної науково-технічної конференції молодих учених та студентів „Актуальні задачі сучасних технологій“, 16-17 листопада 2017 року. — Т. : ТНТУ, 2017. — Том 2. — С. 196. — (Комп'ютерно-інформаційні технології та системи зв'язку).