Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)
Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Дослідження методів оцінки пояснень в інтелектуальних інформаційних системах
(тема)

Виконав:
студент _2_ курсу, групи __ІУСТм-22-1____
__Серафимов Даніїл Ярославович._____
(прізвище, ініціали)

Спеціальність ___122 Комп'ютерні науки__
(код і повна назва спеціальності)

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

Освітня програма __Інформаційні_____
управляючі системи та технології_____
(повна назва освітньої програми)

Керівник __проф. каф. ІУС Сергій ЧАЛИЙ_
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ ___Костянтин ПЕТРОВ_
(підпис) (власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук _____
(повна назва)
Кафедра_____Інформаційних управляючих систем_____
(повна назва)

Рівень вищої освіти _____другий (магістерський)_____
Спеціальність _____122 Комп'ютерні науки_____
(код і повна назва)
Тип програми _____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)
Освітня програма _Інформаційні управляючі системи та технології_____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«20»_____11_____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Серафимову Даніїлу Ярославовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи _____Дослідження методів оцінки пояснень в інтелектуальних_____ інформаційних системах_____
затверджена наказом університету від 16 _____11_____20 23 р. № _1359Ст____
2. Термін подання студентом роботи до екзаменаційної комісії 16 ___01_____ 20 24 р.
3. Вихідні дані до роботи наукові дослідження та статті з методів оцінки пояснень, критерії оцінки пояснень, методи побудови пояснень, методи оцінки пояснень
4. Перелік питань, що потрібно опрацювати в роботі аналіз властивостей інтелектуальних інформаційних систем, дослідження методів побудови пояснень, аналіз методів оцінки пояснень, дослідження критеріїв оцінки пояснень, удосконалення можливісного методу оцінки пояснень, опис інформаційної технології, імплементація інформаційної технології, програмна реалізація технології, експериментальна перевірка методу оцінки пояснень

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	20.11.2023	Виконано
2	Виконання аналізу методів побудови і методів оцінки пояснень	21.11.2023-28.11.2023	Виконано
3	Дослідження критеріїв оцінки пояснень	30.11.2023-05.12.2023	Виконано
4	Удосконалення можливісного методу оцінки пояснень	06.12.2023-11.12.2023	Виконано
5	Опис та імплементація інформаційної технології впровадження удосконаленого методу оцінки пояснень	12.12.2023-15.12.2023	Виконано
6	Програмна реалізація удосконаленого методу оцінки пояснень	16.12.2023-24.12.2023	Виконано
7	Експериментальна перевірка удосконаленого методу оцінки пояснень	25.12.2023-26.12.2023	Виконано
8	Оформлення пояснювальної записки до кваліфікаційної роботи	28.12.2023-06.01.2024	Виконано
9	Надання роботи для перевірки на плагіат	07.01.2024	Виконано
10	Надання роботи на підпис науковому керівникові	10.01.2024	Виконано
11	Попередній захист	11.01.2024	Виконано
12	Надання роботи на рецензію	12.01.2024	Виконано
13	Надання роботи на підпис завідувачу кафедрою	14.01.2024	Виконано
14	Надання підписаної завідувачем кафедрою роботи в ЕК	15.01.2024	Виконано
15	Захист роботи	17.01.2024	Виконано

Дата видачі завдання _20_ ___листопада____20 23 р.

Студент _____
(підпис)

Керівник роботи _____ _____проф. Чалий С. Ф._____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Робота містить: 77 с., 26 рис., 6 табл., 1 додаток, 26 джерел.

ІНФОРМАЦІЙНА ІНТЕЛЕКТУАЛЬНА СИСТЕМА, МАШИННЕ НАВЧАННЯ, МОЖЛИВІСНИЙ МЕТОД ОЦІНКИ ПОЯСНЕНЬ, ОЦІНКА ПОЯСНЕНЬ, ПОЯСНЕННЯ.

У даній роботі виконано аналіз методів оцінки пояснень в контексті інтелектуальних інформаційних систем. Було досліджено властивостей цих систем, їх здатності до генерування пояснень та різноманітним підходам оцінювання цих пояснень. Досліджено існуючі методи побудови пояснень, що включає в себе оцінку їх ефективності. Розглянуто методи побудови пояснень та методи оцінки пояснень. Окрема увага приділена дослідженню критеріїв оцінки пояснень.

В роботі також представлено удосконалення можливісного методу оцінки пояснень, що полягає у проведенні відбору та упорядкуванні за значенням складності всіх коректних пояснень, з відхиленням чутливості не більше заданого значення.

Завершальна частина роботи присвячена програмній реалізації удосконаленого методу оцінки, включаючи експериментальну перевірку його ефективності. Результати дослідження підтверджують значення удосконаленого методу оцінки пояснень у поліпшенні функціональності та користувацького досвіду в інтелектуальних інформаційних системах.

ABSTRACT

The work contains: 77 pages, 26 figures, 6 tables, 1 appendix, 26 sources.

EXPLANATION, EXPLANATION EVALUATION, INFORMATION INTELLIGENT SYSTEM, MACHINE LEARNING, METHOD OF EXPLANATION EVALUATION USING POSSIBILITY THEORY.

This paper analyzes methods for evaluating explanations in the context of intelligent information systems. The properties of these systems, their ability to generate explanations, and various approaches to evaluating these explanations were investigated. The existing methods of building explanations, including the evaluation of their effectiveness, are investigated. Methods of explanation construction and methods of explanation evaluation are considered. Particular attention is paid to the study of criteria for evaluating explanations.

The paper also presents an improvement of a possible method for evaluating explanations, which consists in selecting and organizing all correct explanations by the complexity value, with a sensitivity deviation of no more than a given value.

The final part of the paper is devoted to the programmatic implementation of the improved evaluation method, including experimental verification of its effectiveness. The results of the study confirm the importance of the improved method for evaluating explanations in improving the functionality and user experience in intelligent information systems.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

ІС – інформаційна система

ІІС – інтелектуальна інформаційна система

МН – машинне навчання

ШІ – штучний інтелект

ID – identity document (унікальна ознака об'єкта)

ВСТУП

Штучний інтелект зараз розвивається душе великими темпами, зі зростаючим впливом на різні сфери життя. Цей напрямок характеризується використанням в широкому спектрі застосувань. Однак, ключовим для ефективності та надійності моделей ШІ є глибоке розуміння їхніх механізмів. Критичним аспектом такого розуміння є оцінка пояснень, які відіграють роль у зрозумілості рішень, прийнятих моделлю. Ці пояснення є інструментами для розробки нових моделей, удосконалення існуючих та підвищення довіри користувачів. Оцінці пояснень приділяється велика увага, проводяться детальні дослідження цієї теми.

Мета цієї роботи є дослідження методів оцінки пояснень в інтелектуальних інформаційних системах, дослідження методів для відбору найбільш релевантних пояснень..

Для того, щоб моделі, створені за допомогою технологій штучного інтелекту були ефективними та надійні, важливо розуміти, як вони працюють. Одним із важливих аспектів розуміння таких моделей є оцінка пояснень. Не існує єдиних уніфікованих методів оцінки пояснень. Різні методи використовують різні критерії оцінки, і їхні результати можуть бути суперечливими.

Сучасні ІІС використовують складні алгоритми машинного навчання і тому з точки зору користувача мають вигляд чорного ящика. Непрозорість таких систем знижує довіру користувача до результатів їх роботи. Для того, щоб вирішити цю проблему, використовуються пояснення. Для кожного рішення ІІС можна сформувати декілька пояснень. Для вибору більш релевантного пояснення необхідно виконати їх оцінку.

Не можна виділити уніфіковані методів оцінки пояснень. Різні методи використовують різні критерії оцінки, і їхні результати можуть бути суперечливими.

Розробка нових методів та удосконалення існуючих методів оцінки пояснень, які вирішують ці проблеми, є актуальною задачею.

У роботі зокрема наведено удосконалений метод комплексної оцінки пояснень шляхом відбору пояснень з мінімальною складністю за умов збереження їх чутливості в заданих межах, що дає можливість упорядкувати пояснення за їх складністю і представити найбільш простих пояснень користувачеві.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз властивостей інтелектуальних інформаційних систем

Інтелектуальні інформаційні системи (ІІС) – це комплексні технічні рішення, які використовують методи штучного інтелекту, машинного навчання та аналітики для обробки та аналізу великого обсягу даних з метою прийняття ефективних рішень.

Ці системи можуть автоматизувати процеси збору, обробки, аналізу та використання інформації, що дозволяє вдосконалити різноманітні аспекти діяльності та оптимізувати вирішення завдань. Це дозволяє звільнити час працівників для більш творчих і складних завдань [8].

Основні компоненти інтелектуальних інформаційних систем передбачають застосування передових технологій, таких як штучний інтелект, що дозволяє машинам імітувати людський інтелект, машинне навчання, яке дає змогу системам покращувати свою продуктивність на основі досвіду, та аналітика, яка передбачає вивчення шаблонів даних для отримання значущих висновків.

Синергія технологій надає інтелектуальним інформаційним системам можливість обробляти різноманітні набори даних і робити обґрунтовані прогнози, сприяючи підвищенню ефективності та результативності процесів прийняття рішень.

Ці системи знаходять застосування в різних сферах – від бізнесу і фінансів до охорони здоров’я і виробництва. Автоматизуючи рутинні завдання, аналізуючи величезні масиви даних і надаючи цінну інформацію, інтелектуальні інформаційні системи допомагають організаціям залишатися конкурентоспроможними в сучасному світі, заснованому на даних. Крім того, здатність цих систем адаптуватися і вчитися на новій інформації гарантує, що вони розвиваються разом зі зміною вимог, що робить їх цінним активом для

вирішення складних завдань у різних галузях. Зрештою, інтелектуальні інформаційні системи відіграють ключову роль у підвищенні продуктивності, стимулюванні інновацій та забезпеченні стратегічного розвитку в технологічній галузі, що швидко змінюється.

Інтелектуальні інформаційні системи, що використовують навчання поділяють на ті, що працюють під наглядом та без нагляду, які різні, але можуть доповнювати один одного. При навчанні під наглядом кожна точка даних має заздалегідь визначений результат: класифікована категорія або числове значення. Маючи парні входи і виходи, модель поступово вивчає взаємозв’язки між ними. Завдяки такому навчанню модель стає здатною до прогнозування, з високою точністю, визначаючи майбутні можливості [16].

У разі якщо необхідно провести навчання на даних, які не мають конкретних відповідей і взаємозв’язків між вхідними і вихідними даними. У таких випадках використовуються методи навчання без нагляду, які дозволяють моделі самостійно виявляти закономірності в даних. Таке навчання дає інтелектуальні системи можливість адаптуватися і знаходити зв’язки у неструктурованому, немаркованому наборі даних.

Вибір між цими двома типами моделей залежить від характеру дослідження. Коли пошук вимагає точного передбачення з добре розміченими даними, то використовується навчання під наглядом. Якщо ж дослідження може вимагати вивчення непередбачених закономірностей і розкриття прихованих зв’язків, то використовується навчання без нагляду.

Також методи навчання під наглядом та навчання без нагляду, можуть у деяких випадках використовуватися разом, що відкриває нові можливості.

Системи когнітивного обчислення поділяються на ті, що використовуються для розпізнавання зразків та розпізнавання емоцій.

Розпізнавання шаблонів, або патернів – основа людського сприйняття, яка полягає в основі нашої здатності орієнтуватися у світі. Від розпізнавання облич у натовпі до осягнення синтаксису мови – наш мозок чудово

справляється з вилученням порядку із різної неструктурованої інформація. Ця вроджена навичка спирається на спеціалізовані нейронні мережі, натреновані завдяки впливу патернів, які будують внутрішні моделі, що дають нам змогу класифікувати, передбачати та надавати сенс навколишньому світу. Наприклад, програмне забезпечення для розпізнавання облич використовує цю здатність, детально вивчаючи геометрію рис обличчя, щоб розпізнати особистість.

Однак людська взаємодія виходить за рамки звичайного розрахунку розпізнавання шаблонів. Розпізнавання емоцій, дозволяє людям спілкуватися на більш глибокому рівні. Цей складний процес включає аналіз виразу обличчя, голосових інтонацій і фізіологічних змін, які інтегруються в тонке розуміння емоційного стану. Інтелектуальні інформаційні системи, все частіше використовують розпізнавання емоцій, щоб адаптувати свої реакції до настрою користувача і створити більш природню взаємодію.

Передові моделі включають у свої алгоритми розпізнавання образів міміку і голосові сигнали, що призводить до повніших і тонших інтерпретацій. Це поєднання приховує в собі величезний потенціал і передбачає розвиток когнітивних систем, здатних не тільки розуміти світ, а й відчувати його.

Крім цього інтелектуальні системи можуть використовуватися у абсолютно різних галузях і мати власні особливості відповідно цих галузей, де вони використовуються. Серед них можна виділити ІІС у галузі медицини та фінансів.

У медичній сфері методи штучного інтелекту можуть використовуватися для ретельного аналізу величезних масиви медичних зображень, даних та історій хвороб. Із розвитком технологій алгоритми ІІС могли б розпізнавати тонкі закономірності, невидимі для людського ока, допомагаючи у ранньому виявленні захворювань, виявляти найдрібніші відхилення, передбачаючи можливі захворювання з неймовірною точністю.

Такі досягнення можуть спонукати процеси діагностики виходити за межі інтуїції, спираючись на точність, керовану даними.

Стосовно фінансової сфери, що стрімко розвивається, штучний інтелект дає змогу ухвалювати швидкі й обґрунтовані рішення, можуть виявляти шахрайські дії з безпрецедентною швидкістю і точністю. Крім того, торгові ІІС орієнтуються в складних ринкових ситуаціях з надлюдською спритністю. Їхні відточені алгоритми аналізують велику кількість ринкових даних, виявляють приховані тенденції та передбачають майбутні рухи з вражаючою точністю. Хоча ці досягнення не позбавлені ризику, вони відкривають можливості для оптимізації прибутковості.

Рисунок 1.1 – Класифікація ІІС

1.2 Дослідження методів побудови пояснень

Пояснення – практика пояснення внутрішньої роботи систем штучного інтелекту – є серйозним викликом з технічної точок зору. Останніми роками до пояснень спостерігається сплеск інтересу з боку різних галузей. Незважаючи на те, що було розроблено багато методів пояснення, залишаються питання про те, як найкраще донести пояснення користувачам. Прагнення до прозорості та інтерпретованості набуло першочергового значення.

У актуальних системах і додатках, особливо тих, що передбачають прийняття відповідальних рішень, здатність розуміти і довіряти рішенням, прийнятим моделями ШІ, має фундаментальне значення. Два основні підходи, які часто називають прозорими (біла скринька) і чорними (чорна скринька) моделями, окреслюють спектр інтерпретованості в цій сфері [15].

Прозорі моделі, втілені у вигляді дерев рішень, лінійної регресії та систем, заснованих на правилах, пропонують чіткий погляд на процес прийняття рішень. Їх простота сприяє зрозумілості, дозволяючи користувачам простежити логіку і коефіцієнти, тим самим вселяючи довіру. Однак компроміс полягає в тому, що вони потенційно обмежені у відображенні складності, притаманної певним наборам даних, що перешкоджає їхній прогностичній здатності у складних сценаріях.

З іншої сторони, існують моделі "чорної скриньки", прикладом яких є глибокі нейронні мережі. Ці моделі чудово вловлюють складні закономірності та нелінійні взаємозв'язки, демонструючи чудову прогностичну ефективність. Проте їхнім недоліком є непрозорість процесів прийняття рішень. Користувачі часто стикаються з відсутністю розуміння того, чому було прийнято те чи інше рішення, що ставить під сумнів довіру та підзвітність.

Порівняння двох основних категорій моделей наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння прозорих та непрозорих моделей прийняття рішень в ІІС

	Прозорі	Непрозорі
Прозорість	Висока прозорість; внутрішня логіка зрозуміла і доступна	Низька прозорість; внутрішні процеси важко зрозуміти
Інтерпретованість	Вища інтерпретованість та пояснюваність	Нижча інтерпретованість та пояснюваність
Розуміння користувачами	Користувачі можуть переглянути внутрішню логіку та зрозуміти рішення	Користувачі можуть не мати детального уявлення про те, як приймаються рішення
Застосування	Не підходить для повсякденного використання через нижчу точність	Вища точність, але непрактична для критичних застосувань через відсутність прозорості
Практичність	Обмежене практичне використання в повсякденних сценаріях через нижчу точність	Непрактично для критичних програм через брак розуміння

Прозорі моделі – це моделі, які за своєю суттю можуть бути інтерпретовані користувачами. Отже, найпростіший спосіб досягти інтерпретованості – це використовувати алгоритми, які створюють інтерпретовані моделі, такі як дерева рішень, прості моделі найближчого сусіда або лінійна регресія. Однак найефективніші моделі часто не піддаються

інтерпретації або піддаються частковій інтерпретації. Забезпечення високої точності моделі при збереженні достатнього рівня її зрозумілості є постійним викликом. Методи діагностичної інтерпретації моделей пропонують відокремити пояснення від моделі МН.

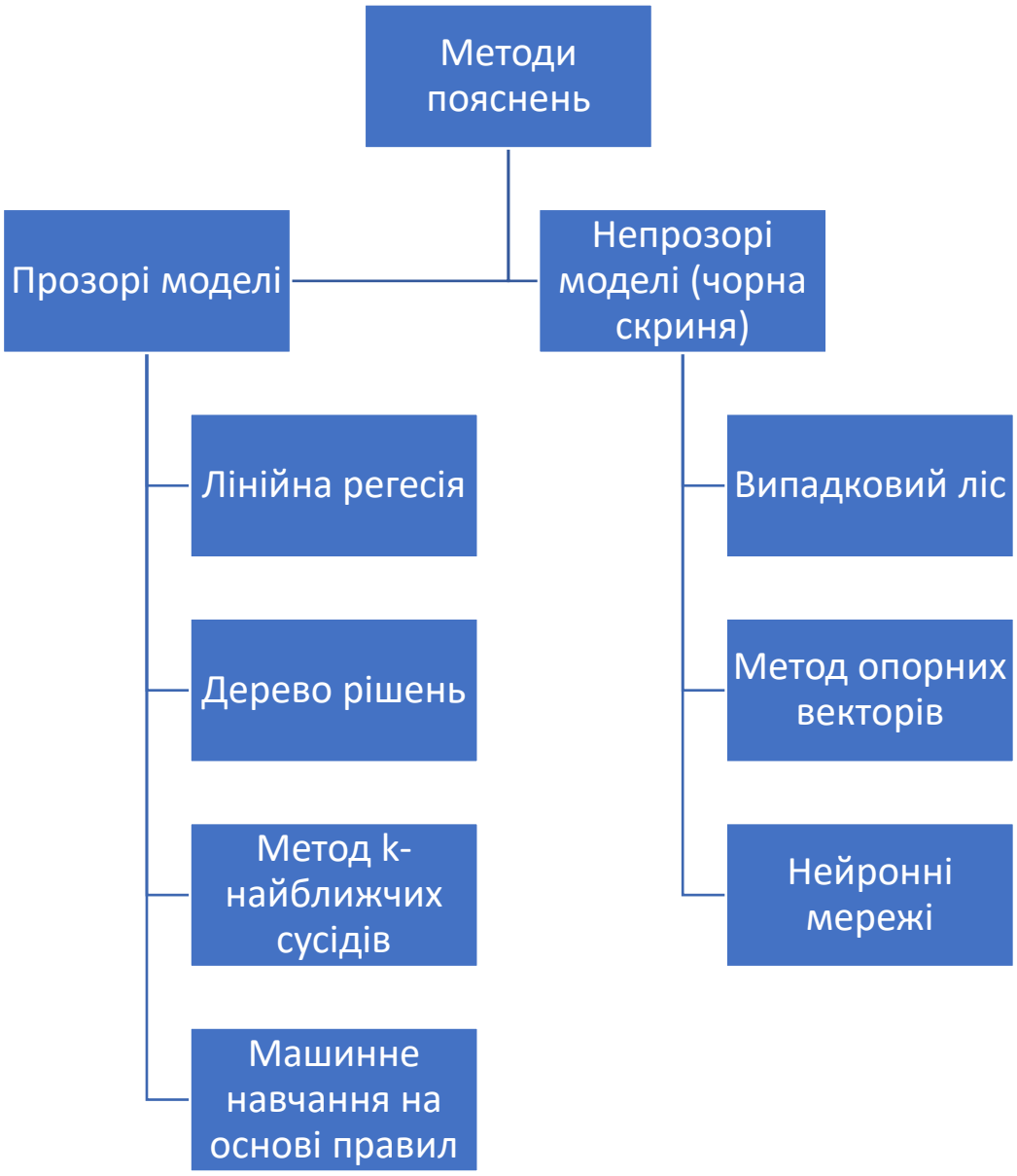

Рисунок 1.2 – Моделі та методи формування рішень в інтелектуальних систем

Лінійна/логістична регресія. Моделі лінійної та логістичної регресії широко використовуються в прогнозній аналітиці для моделювання зв'язку між залежною змінною та однією або кількома незалежними змінними. Лінійна регресія використовується для неперервних залежних змінних, тоді як логістична регресія використовується для задач бінарної класифікації. Ці моделі припускають лінійний зв'язок між передбачувальними і прогнозованими змінними, що робить їх відносно простими і зрозумілими для інтерпретації. Пояснюваність цих моделей безпосередньо пов'язана зі зрозумілістю змінних, і вони відповідають вимірам прозорості, визначеним у рамках XAI [17]. Хоча ці моделі вважаються прозорими, може виникнути потреба у постфактум поясненні з використанням візуальних методів для покращення їхньої інтерпретації. Моделі лінійної та логістичної регресії продемонстрували високий рівень точності в різних сферах застосування, що підвищує їхню актуальність і важливість у процесах прийняття рішень.

Дерева рішень – це моделі, які підтримують класифікацію та регресію і часто називаються деревами класифікації та регресії. У своїй найпростішій формі дерева рішень піддаються моделюванню та декомпозиції, що робить їх зрозумілими та керованими для людини. Однак, коли дерева стають більшими, вони можуть вимагати алгоритмічної прозорості через складні зв'язки між ознаками.

K-nearest neighbors – це непараметричний класифікатор з керованим навчанням, який використовується для класифікації та прогнозування на основі близькості до інших точок даних. Він не робить припущень щодо вихідних даних і може бути адаптований до конкретних проблем, що забезпечує пояснюваність моделі. K-NN моделі легко пояснити та інтерпретувати, що робить їх широко застосовними в різних галузях.

Моделі, що навчаються на основі правил – це моделі, які генерують правила для опису даних, на яких вони мають намір навчатися. Ці правила можуть мати форму умовних правил "якщо-тоді" або більш складних

комбінацій. Системи на основі нечітких правил – це тип систем, що навчаються на основі правил, які використовують нечіткі множини для вираження знань у вигляді набору нечітких правил для вирішення складних проблем реального світу, де існують невизначеність, неточність і нелінійність. Системи, що навчаються на основі правил, як правило, можуть бути інтерпретовані користувачами, але основним недоліком є кількість і довжина згенерованих правил, що може знизити інтерпретованість.

Загальні адитивні моделі – це лінійні моделі, в яких результат є лінійною комбінацією функцій на основі вхідних даних. Структура легко інтерпретується, оскільки дозволяє користувачеві перевірити важливість кожної змінної і те, як вона впливає на очікуваний результат. Моделі використовуються в різних галузях, для визначення ризиків та оцінки продуктивності. Методи візуалізації, такі як графіки залежностей, зазвичай використовуються для покращення інтерпретації моделей. Однак взаємодії можуть стати занадто складними для моделювання, що вимагає застосування методів декомпозиції для аналізу моделі.

Баєсові моделі – тип імовірнісних моделей, які використовують байєсівський висновок для оновлення ймовірностей на основі нових даних. Вони моделюють статистичні зв'язки між змінними і можуть використовуватися як для регресії, так і для класифікації. Байєсівські моделі є гнучкими і можуть обробляти складні структури даних, що робить їх корисними в різних галузях, включаючи фінанси, охорону здоров'я та енергетику. Однак, вони можуть бути обчислювально інтенсивними і вимагають попередніх знань або припущень про дані. Байєсівські моделі, як правило, піддаються інтерпретації, але їхня складність може вимагати використання математичних і статистичних інструментів для аналізу.

Таблиця 1.2 – Прозорі методи

Лінійна/логістична регресія	Широко використовуються в передбачувальній аналітиці для моделювання зв'язку між залежною змінною та однією або кількома незалежними змінними. Ці моделі вважаються прозорими та інтерпретованими, що робить їх актуальними в різних сферах застосування.
Дерева рішень	Підтримують класифікацію та регресію, пропонуючи прозорість та інтерпретованість. У своїй найпростішій формі дерева рішень піддаються моделюванню та декомпозиції, що робить їх зрозумілими та керованими для людини.
K-nearest neighbors	Непараметричний класифікатор з керованим навчанням, який використовується для класифікації та прогнозування на основі близькості до інших точок даних. Легко пояснити та інтерпретувати, що робить його широко застосовним у різних галузях.
Навчаються на основі правил	Можна інтерпретувати, системи на основі нечітких правил використовують нечіткі множини для вирішення складних реальних проблем. Кількість і довжина згенерованих правил може впливати на їхню інтерпретацію.
Загальні адитивні моделі	Лінійні моделі, в яких результат є комбінацією функцій на основі вхідних даних, піддаються інтерпретації, часто візуалізуються за допомогою графіків залежностей.

Кінець таблиці 1.2

Баєсові моделі	Використовують байєсівський висновок для моделювання статистичних взаємозв'язків між змінними, пропонуючи гнучкість та інтерпретованість, але вимагаючи обчислювальних ресурсів та попередніх знань.

Випадковий ліс – модель, яка підвищує точність одиночних дерев рішень шляхом об'єднання декількох дерев для зменшення дисперсії та покращення узагальнення. Вона широко використовується в різних галузях, для таких завдань, як розпізнавання діяльності та прогнозування витрат. Однак ансамблева природа випадкового лісу ускладнює його інтерпретацію, часто вимагаючи пояснення постфактум з використанням локальних пояснень.

Метод опорних векторів – алгоритм керованого навчання, який використовується для задач класифікації та регресії. Вимагає постфактум пояснень через свою складність, часто потребуючи спрощення, локальних пояснень, візуалізації та пояснень на прикладах.

Багатошарова нейронна мережа – нейронна мережа прямого поширення, яка складається з декількох шарів взаємопов'язаних вузлів, кожен з яких виконує певну функцію. Вхідний шар отримує вхідні дані, а вихідний шар виробляє вихід.

Приховані шари між вхідним і вихідним шарами виконують складні обчислення для перетворення вхідних даних у бажаний результат. Алгоритми схильні до проблеми зникаючого градієнта, яка виникає, коли глибока багатошарова мережа прямого поширення не може поширювати корисну інформацію про градієнт від вихідного кінця моделі назад до шарів біля її входу. Ця проблема може призвести до низької продуктивності та повільної збіжності моделі. Є потужним інструментом, але його пояснюваність є проблемою.

Згорткова нейронна мережа – тип глибокої нейронної мережі, яка широко використовується в програмах комп'ютерного зору. Призначені для автоматичного вивчення просторових ієрархій ознак шляхом зворотного поширення з використанням, таких блоків як згортка, об'єднання і повністю з'єднані шари. Структура CNN є складною, з внутрішніми зв'язками, які важко пояснити.

Щоб вирішити проблему пояснюваності, дослідники розробили різні методи, такі як розуміння процесу прийняття рішень шляхом відображення вихідних даних у просторі вхідних даних, щоб визначити частини вхідних даних, які є дискримінаційними для вихідних даних, і заглиблення всередину мережі для інтерпретації того, як проміжні шари бачать зовнішнє середовище. Крім того, дослідники використовували методи пояснення постфактум.

Рекурентна нейронна мережа – тип штучної нейронної мережі, який широко використовується в обробці природної мови та аналізі часових рядів. Призначені для обробки послідовних даних, підтримуючи прихований стан, який фіксує контекст попередніх вхідних даних. Цей прихований стан оновлюється на кожному часовому кроці, що дозволяє мережі фіксувати довгострокові залежності в даних.

Для кожного з них існують специфічні підходи та методи, що спрямовані на вирішення проблеми інтерпретації та пояснення. Ці методи включають використання постфактум пояснень, локальних аналізів, візуалізацій та інших технік.

Таблиця 1.3 – Непрозорі методи

Випадковий ліс	Об'єднує кілька дерев для підвищення точності та узагальнення. Використовується в різних додатках, але може бути складним для інтерпретації, часто вимагаючи пояснення постфактум.
Метод опорних векторів	Алгоритм керованого навчання, який використовується для задач класифікації та регресії, часто вимагаючи пояснень постфактум через свою складність.
Багатошарова нейронна мережа	Потужний інструмент, але його пояснюваність є проблемою.
Згорткова нейронна мережа	Широко використовуються в комп'ютерному зорі, застосовуючи такі шари, як згортка та об'єднання, для автоматичного вивчення просторових ієрархій ознак.
Рекурентна нейронна мережа	Використовуються в обробці природної мови та аналізі часових рядів, Незважаючи на зусилля, спрямовані на розуміння процесу навчання та прийняття рішень, інтерпретованість моделі залишається складним завданням, що спонукає використовувати постфактум методи.

У сучасних моделях машинного навчання існує різноманіття методів пояснення, кожен з яких відіграє важливу роль у забезпеченні прозорості та зрозумілості для кінцевих користувачів. Текстові пояснення є ефективним способом ілюстрування процесу прийняття рішень, особливо в сфері обробки природної мови, де вхідні дані представлені у текстовій формі. Візуальні пояснення, з іншого боку, використовують графічні зображення для

зрозумілого представлення внутрішньої роботи моделей. Локальні пояснення допомагають користувачам зрозуміти конкретні прогнози або рішення моделі, в той час як пояснення на прикладах демонструють механізм роботи моделі через конкретні інстанції. Пояснення шляхом спрощення зменшують складність моделей, роблячи їх зрозумілими для ширшої аудиторії, тоді як метод пояснення релевантності ознак зосереджується на визначенні ключових вхідних ознак, які впливають на рішення моделі.

Кожен із цих підходів відіграє унікальну роль у роз'ясненні внутрішніх механізмів моделей машинного навчання, дозволяючи користувачам не тільки зрозуміти, але й довіряти результатам, які ці моделі надають. Зрештою, ці методи пояснення сприяють більшій прозорості та прийнятності штучного інтелекту в широкому колі застосувань.

Таблиця 1.4 – Види пояснень в ІІС

Текстові пояснення	Тип техніки пояснень, щоб дати уявлення про процес прийняття рішень моделями машинного навчання. Текстові пояснення особливо корисні в програмах обробки природної мови, де вхідні дані мають форму тексту.
Візуальні пояснення	Візуальні пояснення використовують графічні та візуальні зображення, щоб прояснити внутрішню роботу складних моделей, полегшуючи користувачам розуміння і довіру до результатів роботи моделі.
Локальні пояснення	На відміну від глобальних пояснень, які мають на меті пояснити загальну поведінку моделі, локальні пояснення призначені для з'ясування обґрунтування конкретних прогнозів або рішень, прийнятих моделлю для конкретного прикладу вхідних даних.

Кінець таблиці 1.4

Пояснення на прикладах	Метод передбачає використання конкретних прикладів для демонстрації того, як модель приходить до своїх прогнозів або класифікацій, забезпечуючи тим самим прозоре і зрозуміле розуміння її внутрішньої роботи.
Пояснення шляхом спрощення	Метод передбачає зменшення складності внутрішньої роботи моделі, щоб зробити її більш доступною для неспеціалістів, тим самим сприяючи глибшому розумінню поведінки моделі.
Пояснення релевантності ознак	Має на меті надати інтерпретоване розуміння процесу прийняття рішень моделями машинного навчання шляхом визначення вхідних ознак, які є найбільш релевантними для прогнозів моделі. Цей метод передбачає кількісну оцінку впливу кожної вхідної змінної на вихід моделі, що проливає світло на фактори, які впливають на конкретні результати.

Сучасні інтелектуальні системи використовують методи машинного навчання для побудови моделей, що забезпечують прийняття рішень. Ці методи зазвичай є непрозорими для користувача або можуть бути недоступними з юридичних причин (захист авторського права). Тому непрозорість прийняття рішень в таких системах знижує довіру користувача до процесу прийняття рішень, що приводить до неефективного використання таких рішень. Зазначене свідчить про важливість побудови пояснень в інтелектуальних системах.

1.3 Аналіз методів оцінки пояснень

У той час як існують стандартні метрики оцінювання для оцінки ефективності моделі, не існує узгодженої стратегії оцінювання пояснювань інтелектуальних інформаційних систем. Як наслідок, загальна стратегія оцінювання полягає в тому, щоб показати окремі, потенційно відібрані приклади, які виглядають обґрунтованими і проходять первісну перевірку “на око”. Відсутність кількісної оцінки пояснень може перешкоджати коректному дослідженню інтерпретованості.

Також оцінка правдоподібності та переконливості пояснення для людей відрізняється від оцінки його правильності, і ці критерії оцінки не слід змішувати. Стверджується, що не гарантується, що правдоподібне пояснення також правдиво відображає міркування моделі. Але думка, що виключення людини з процесу оцінювання робить його більш справедливим і вірним власному погляду класифікатора на проблему, ніж представлення людського погляду, теж має своє підґрунтя. Це пояснюють тим, що нерозумне на вигляд пояснення може вказувати або на помилку в міркуваннях прогностичної моделі, або на помилку в методі отримання пояснення. Візуальна перевірка правдоподібності пояснення, не може зробити це розрізнення. Пояснення може сприйматися як помилкове, хоча воно правдиво відображає міркування моделі, а саме модель є помилковою або некоректною. Це може бути основним недолік при оцінці пояснень, адже перевірка того, чи пояснення виглядає коректним, оцінює лише точність моделі "чорного ящика", але не оцінює вірність пояснення. Хоча правильність пояснення не залежить від правильності прогнозу, візуальний огляд не може розрізнити їх.

Згадавши про дві основні категорії моделей прийняття рішень – прозорі та непрозорі, важливо визначити, що надійність та ефективність цих моделей в значній мірі залежать від здатності користувача розуміти їхні внутрішні процеси.

Для досягнення цього розуміння важливим є використання методів оцінки пояснень. Як було визначено до цього, довіра до прийняття рішень є ключовим аспектом, тому необхідно мати точне уявлення про оцінки пояснювальних здатностей моделей та їхню релевантність для забезпечення впевненості користувачів у прийнятих рішеннях.

Однією з актуальних проблем у сфері пояснень є відсутність фіксованого визначення того, яким має бути пояснення. Ця проблема зумовлена суб'єктивною сутністю пояснень, різноманітністю ситуацій, в яких вони використовуються, а також численними і потенційно суперечливими цілями (простота, вірність, повнота), які вони можуть переслідувати в різних контекстах. Зокрема, цей контекст включає характер цільової аудиторії пояснення (фахівець з даних, нефахівець, експерт у прикладній галузі або аудитор). Тому створені методи, як правило, оцінюються більше якісно, ніж кількісно. Однак дослідження про те, як оцінювати методи пояснень, розвиваються та різні науковці можуть надавати різні категорії і ознаки методів оцінки пояснень. Проте, в основному можна такі основні категорії:

- методи, які використовують суто технічні оцінки, об’єктивні: ми перевіряємо такі властивості пояснень, як різноманітність відповідей або їх складність;

- методи, використовують людське оцінювання, або за допомогою простих тестових завдань, або в реальних умовах.

Перші дозволяють проводити кількісну оцінку але можуть бути відірвані від фундаментальних цілей пояснень, якщо не оцінюється релевантність тестових завдань. Крім того, ці оцінки, як правило, адаптовані до типу оцінюваного пояснення і не дозволяють порівнювати пояснення різної природи.

Людські, або суб’єктивні, оцінки пояснень діляться на два основних підходи. До першого відносяться ті, що базуються на суб'єктивних вимірах (користувача просять оцінити "зрозумілість" або різні критерії, пов'язані з його

відчуттями). До других відносяться ті, що вивчають систему людина-ШІ ззовні, вимірюючи час реакції суб'єкта або точність прийнятих рішень, а також порівнюючи використання пояснень з різними базовими рівнями.

Коректність є фундаментальною властивістю в оцінці пояснень, що стосується правдивості та вірності пояснення по відношенню до пояснюваної прогностичної моделі. Він зосереджується на тому, наскільки точно пояснення відображає поведінку базової моделі чорного ящика, підкреслюючи описову точність, а не прогностичну точність. Мета полягає в тому, щоб пояснення було "нічим іншим, як правдою", що вказує на високий рівень вірності поведінки моделі чорної скриньки.

Повнота в оцінці пояснень, зосереджується на тому, якою мірою пояснення описує поведінку прогностичної моделі, яку воно прояснює. Висока повнота бажана для того, щоб забезпечити достатню деталізацію пояснення при збереженні балансу між коректністю і компактністю.

Узгодженість відображає, якою мірою ідентичні вхідні дані дають ідентичні пояснення. Ця властивість стосується детермінізму методу пояснення, гарантуючи, що на пояснення не впливають випадкові фактори або деталі реалізації. Крім того, для методів пояснення, які не розглядають внутрішню частину чорної скриньки, а лише спостерігають вхідні і вихідні дані, узгодженість стосується інваріантності реалізації, яка стверджує, що дві моделі, які дають однакові результати для всіх вхідних даних, повинні мати однакові пояснення.

Безперервність розглядає, наскільки безперервною є функція пояснення. Безперервна функція гарантує, що невеликі зміни вхідних даних, для яких реакція моделі майже ідентична, не призведуть до значних змін у поясненні. Безперервність також додає узагальнюваності за межами конкретних вхідних даних або узагальнюваності в нових контекстах.

Контрастність оцінює невеликі, незначні розбіжності в різних поясненнях і має на меті полегшити порівняння з іншими цілями або подіями.

Коваріантна складність враховує складність коваріацій, що використовуються в поясненні, з точки зору семантичного значення та взаємодії між коваріатами та об'єктом. Коваріати в поясненні повинні бути зрозумілими, а концепції повинні мати безпосередню інтерпретацію, зрозумілу людині.

Компактність враховує розмір пояснення і мотивується обмеженнями людської когнітивної здатності. Пояснення мають бути небагатослівними, короткими і не надлишковими, щоб уникнути надмірного обсягу пояснення.

Композиція описує формат подання, організацію та структуру пояснення, також спосіб, у який пояснення подається користувачеві, підвищує його зрозумілість. Деякі формати подання зазвичай вважаються такими, що легше інтерпретуються, ніж інші. Отже, ця властивість стосується того, як щось пояснюється, а не того, що пояснюється.

Упевненість стосується того, чи є в поясненні міра впевненості або інша ймовірнісна інформація. Вона може відображати два аспекти впевненості: міру впевненості у передбаченні чорної скриньки або правдивість/ймовірність пояснення. Стосовно останнього аспекту є сумніви, що посилання на ймовірності може бути не настільки ефективним, оскільки людям важко правильно оцінити ймовірності.

Контекст – це ступінь урахування потреб користувача при складанні зрозумілих пояснень. Пояснення мають відповідати потребам і рівню знань користувача. Крім того існує поширена думка, що пояснення мають використовуватися і бути зрозумілими не тільки спеціалістам сфери ШІ, а й цілій низці зацікавлених сторін, наприклад, звичайним споживачам.

Когерентність оцінює, наскільки пояснення узгоджується з відповідними побічними знаннями, переконаннями та загальним консенсусом.

Контрольованість показує, якою мірою користувач може контролювати, коригувати або взаємодіяти з поясненням.

Оцінка пояснень є складним завданням, оскільки вимагає урахування різноманітних факторів, і може залежати від специфіки моделей, які використовується для побудови рішення та пояснень.

Описані вимоги не завжди можуть задовольняти всі потреби або можуть не відповідати початковим цілям, які закладалися для конкретних моделей. Правильна оцінка пояснень має велике значення, оскільки вона дозволяє виявити їхні недоліки, порівняти різні варіанти та покращити їх, усуваючи виявлені недоліки. Здійснення такої оцінки може відбуватися за різними критеріями в залежності від контексту використання пояснень [24].

Таблиця 1.5 – Вимоги до пояснень в ІІС

Коректність	Наскільки правдивими є пояснення порівняно з "істинною" поведінкою "чорної скриньки".
Повнота	Наскільки поведінка чорної скриньки описана в поясненні.
Узгодженість	Наскільки детермінованим та інваріантним до реалізації є метод пояснення
Безперервність	Наскільки безперервною та узагальнюючою є функція пояснення
Контрастність	Описує розбіжності в різних поясненнях
Коваріантна складність	Наскільки складними є ознаки в поясненні.
Компактність	Розмір пояснення
Композиція	Описує формат представлення та організацію пояснення
Упевненість	Описує наявність і точність інформації про ймовірність у поясненні

Кінець таблиці 1.5

Контекст	Наскільки пояснення відповідає користувачеві та його потребам.
Когерентність	Описує, наскільки пояснення узгоджується з попередніми знаннями та переконаннями
Контрольованість	Описує, наскільки інтерактивним або керованим є пояснення для користувача.

Якщо інтелектуальна система для прийняття рішень представляє моделлю чорного ящика, то при створенні пояснень будується окрема система пояснень, яка базується на спрощеній моделі процесу прийняття рішень. У цій моделі використовуються вхідні та вихідні дані інтелектуальної системи, а в разі наявності можливості – також і проміжні дані. На цій основі будується пояснення, яке служить для розкриття процесів, що відбуваються всередині системи чорного ящика. Ця пояснювальна система дозволяє користувачам або іншим зацікавленим сторонам краще зрозуміти, як система приймає рішення, використовуючи доступні дані та прогнози.

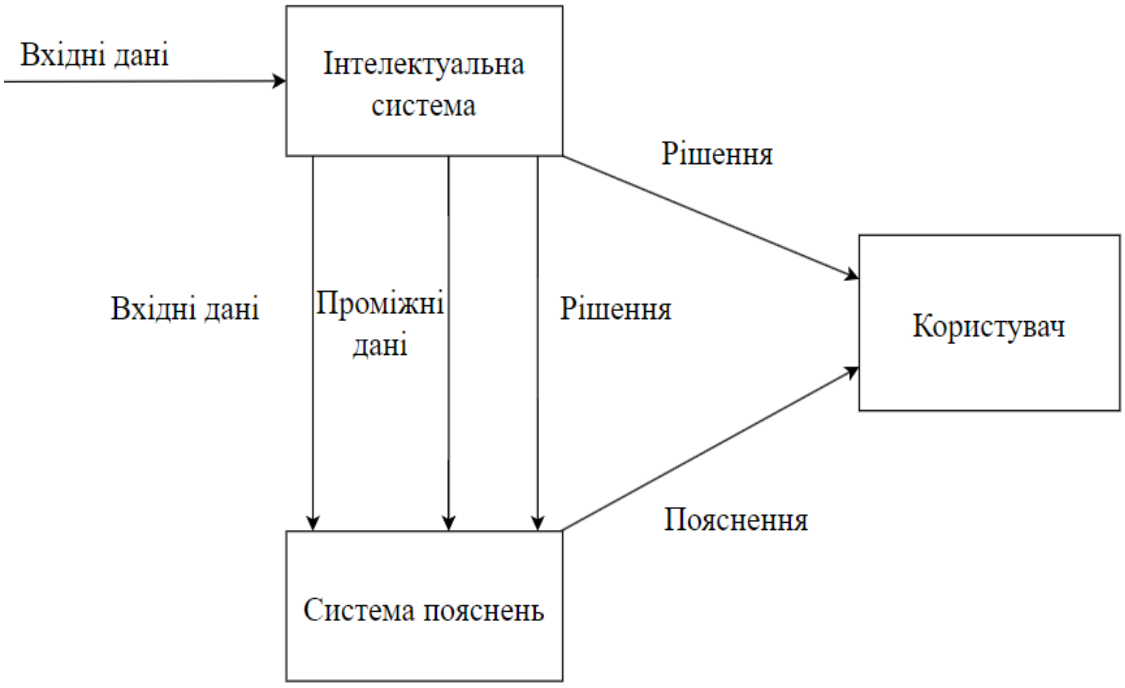

Рисунок 1.3 – Схема взаємодії інтелектуальної системи та системи пояснень

1.4 Постановка задачі дослідження

Об’єктом дослідження є процес оцінки пояснень у інтелектуальних інформаційних системах.

Предметом дослідження виступають методи оцінки таких пояснень.

Метою роботи є дослідження методів оцінки пояснень у ІІС з метою відбору найбільш релевантних пояснень.

Науковою новизна полягає в удосконаленні метод комплексної оцінки пояснень. Це передбачає відбір пояснень з мінімальною складністю при збереженні заданої межі їх чутливості. Такий підхід дозволить упорядкувати

пояснення за рівнем складності та надати користувачеві найбільш прості та зрозумілі пояснення.

Задачі роботи:

- аналіз методів побудови пояснень в інтелектуальних інформацій них системах;

- детальний аналіз існуючих методів оцінки пояснень;

- удосконалення методу оцінки пояснень, що включатиме вибір найбільш релевантних та простих пояснень;

- розробка інформаційної технології для комплексної оцінки пояснень, що дозволить ефективно застосовувати удосконалений метод;

- програмна реалізація результатів теоретичних досліджень для практичного використання в інтелектуальних інформаційних системах.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ТА КРИТЕРІЇВ ОЦІНКИ ПОЯСНЕНЬ

2.1 Дослідження критеріїв оцінки пояснень

У сфері машинного навчання складні моделі "чорних скриньок" слугують потужними інструментами для вирішення складних проблем. Привабливість цих моделей має вирішальне значення для їхнього успішного застосування, гарантуючи, що зацікавлені сторони зможуть зрозуміти їхню внутрішню роботу і довіряти їхнім результатам. В основі цих моделей лежать вхідні дані, тобто змінні, які визначають прогнози. Крім того, обговорюються будь-які кроки попередньої обробки вхідних даних, що підвищує прозорість. Процес навчання, критичний етап у розробці моделі, передбачає перетворення необроблених даних на інструмент прогнозування.

Архітектура моделі є фундаментальним аспектом, який заслуговує на увагу. Розуміння використовуваних шарів, вузлів та унікальних структур забезпечує контекст для процесу прийняття рішень в моделі. Пояснення описує, чому було обрано певну архітектуру, висвітлюючи її переваги для конкретної проблеми, що розглядається.

Загальним уявленням про якість пояснень є кількісна оцінка ступеня, який відображає, як змінюється сама функція прогнозування у відповідь на значущі викривлення.

Пояснення щодо чутливості моделі є ще одним ключовим компонентом. Чутливість визначається як здатність моделі реагувати на зміни вхідних даних. Оцінка чутливості включає визначення ступеня, в якому пояснення змінюється у відповідь на незначні модифікації вхідних даних. Це допомагає зрозуміти, наскільки стабільні та надійні рішення моделі у різних умовах. Отже, Чутливість задається через відхилення співвідношень вхідних даних та рішення за умови схожості пояснень [5].

Складність пояснень відноситься до їх здатності бути зрозумілими та доступними для користувача. Це означає, що пояснення повинні бути

представлені з мінімальною складністю та з урахуванням обмежень сприйняття людини-користувача . Складність пояснень є важливою для забезпечення їх зрозумілості та доступності для користувача, що дозволяє забезпечити ефективну комунікацію між інтелектуальною системою та користувачем .

Точність. Для оцінки точності пояснень можуть використовуватися різноманітні методи, включаючи порівняння пояснень з реальними даними та визначення ступеня відповідності між ними, а також врахування можливих відхилень та похибок . Точність пояснень є важливим критерієм оцінки, оскільки вона визначає, наскільки коректно та адекватно пояснення відображає реальний процес роботи інтелектуальної системи.

Таблиця 2.1 – Основні критерії оцінки пояснень

Критерій	Опис
Чутливість	Визначає, наскільки зміни вхідних даних можуть впливати на пояснення. Модель повинна бути чутливою до змін в даних, щоб можна було ефективно пояснити причини вибору конкретного рішення.
Складність	Це ступінь складності моделі, яка використовується для прийняття рішень. Зазвичай, пояснювальні моделі повинні бути простішими і зрозумілішими, ніж моделі, які вони пояснюють. Забезпечення адекватної складності є ключовим аспектом в забезпеченні доступності пояснень для користувачів.
Точність	Вказує, наскільки точно вона може пояснити прийняті рішення моделі машинного навчання. Точність важлива, оскільки неправильні або недостовірні пояснення можуть призвести до неправильного розуміння системи користувачем.

З огляду на ці критерії, важливо розробляти методи оцінки пояснень, які можуть ефективно виміряти ці аспекти. Такі методи можуть включати, наприклад, автоматизовані інструменти для оцінки точності, чутливості та складності пояснень, а також експертні оцінки для гарантії їх зрозумілості та релевантності. Використання цих критеріїв у процесі розробки та оцінки моделей машинного навчання допомагає забезпечити, що кінцеві рішення моделі є не тільки точними, але й прозорими та зрозумілими для користувачів.

Традиційно, щоб виміряти чутливість, використовується підхід, заснований на вивченні змін пояснення у відповідь на мінімальні зміни вхідних даних. Таким чином, чутливість пояснення визначається через оцінку варіацій у відповіді функції на незначні коливання вхідних даних. Цей процес включає аналіз того, наскільки значно змінюється пояснення при мінімальних модифікаціях вхідних даних. Для здійснення цього аналізу, часто використовується методика, що дозволяє оцінити загальну зміну чутливості пояснення.

Існує також підхід, який зосереджується на оцінці максимальної чутливості пояснення. Цей метод передбачає визначення найбільшого ступеня зміни пояснення при невеликих змінах вхідних даних. Така оцінка важлива для розуміння меж чутливості пояснення та його поведінки під час коливань даних. Оцінка максимальної чутливості може бути здійснена за допомогою методів Монте-Карло, які дозволяють надійно виміряти цю характеристику.

Важливим аспектом у вивченні чутливості є розуміння того, що надмірно чутливі пояснення можуть бути менш корисними, оскільки вони можуть надавати занадто загальну або нестабільну інформацію. Однак, важливо зазначити, що зниження чутливості пояснення до мінімуму також може бути не завжди доцільним. Оптимальне пояснення повинно забезпечувати баланс між точністю та стабільністю, а занадто низька чутливість може призвести до того, що пояснення стануть надмірно спрощеними або нерелевантними.

Важливо розуміти, що чутливість пояснення не завжди слід розглядати як негативний аспект. У деяких випадках, певний рівень чутливості може бути цілком обґрунтований, наприклад, якщо модель природно чутлива до певних видів даних, або якщо пояснення спеціально розроблялися для виявлення тонких відмінностей у поведінці функції. У цих випадках, зменшення чутливості може призвести до втрати важливої інформації, яка важлива для розуміння поведінки моделі.

Тому, коли мова йде про зниження чутливості пояснення, важливо зважати на потенційні переваги та недоліки. Надмірна зосередженість на мінімізації чутливості може призвести до того, що пояснення стануть занадто загальними або неінформативними, тоді як певний рівень чутливості може бути корисним для забезпечення глибшого розуміння моделі та її поведінки.

Таким чином, підхід до вимірювання та регулювання чутливості пояснення вимагає обережності та розуміння контексту, у якому вони використовуються. Важливо знайти оптимальний баланс, який дозволить зберегти релевантність та корисність пояснення, уникаючи при цьому надмірної варіативності або невизначеності.

2.2 Удосконалення можливісного методу оцінки пояснень

Можливісний метод зосереджується на визначенні відповідності пояснень процесів прийняття рішень у інтелектуальних системах, враховуючи взаємозв'язки між вхідними даними та рішеннями, прийнятими системою [4]. Цей метод включає аналіз чутливості, точності та складності пояснень шляхом порівняння значень та обсягу даних, використаних у поясненні.

На першому етапі методу оцінюється точність пояснення. У контексті можливісного підходу, точність вимірюється наскільки необхідним було прийняття конкретного рішення, заснованого на заданих вхідних даних формули (2.1).

$$C\big(E(X_i)\big) = true \mid 1 - \Pi(X \backslash X_i) > 0.5. \quad (2.1)$$

де $1 - \Pi(X \backslash X_i)$ – можливість вибору інших пояснень.

На другому етапі методу відбувається оцінка складності реалізації та застосування пояснення. Складність конкретного пояснення $E(X_i)$ визначається через кількість використаних вхідних змінних $|D_i|$ для цього пояснення.

На третьому етапі проводиться аналіз чутливості пояснення. Чутливість визначається як відхилення відносин між вхідними даними та прийнятим рішенням при умові, що пояснення залишаються схожими. У контексті можливісного підходу, чутливість SN визначається через відмінності у можливостях використання різних наборів вхідних даних для досягнення тієї ж конкретної пояснювальної відповіді при однаковому кінцевому результаті формула (2.2).

$$SN(E) = max\big(\big|\Pi(D_i) - \Pi\big(D_j\big)\big|\big) \quad (2.2)$$

де $\Pi(D_i) - \Pi\big(D_j\big)$ відмінність можливостей використання різних вхідних даних для досягнення одного й того ж пояснення при отриманні однакового результату.

У ситуації, коли подібні результати призводять до однакових пояснень, важливо враховувати відповідність між можливостями, які надають вхідні дані, та прийнятими рішеннями формули (2.3).

$$SN(E) = \left(\left|\frac{\Pi(D_i)}{\Pi(X_i)} - \frac{\Pi\big(D_j\big)}{\Pi\big(X_j\big)}\right|\right) | E_i = E_j \quad (2.3)$$

У можливісній оцінці формула (2.3) закладена концепція, що чутливість представляє максимальне відхилення між потенціалами впливу вхідних даних на прийняті рішення системою штучного інтелекту.

На етапі 4 проводиться відбір та упорядкування за значенням складності всіх коректних $E(D_i)$ пояснень, з відхиленням чутливості $SN(E)$ не більше n.

Послідовність етапів удосконаленого можливісного методу оцінки пояснень наочно представлено на рисунку 2.1.

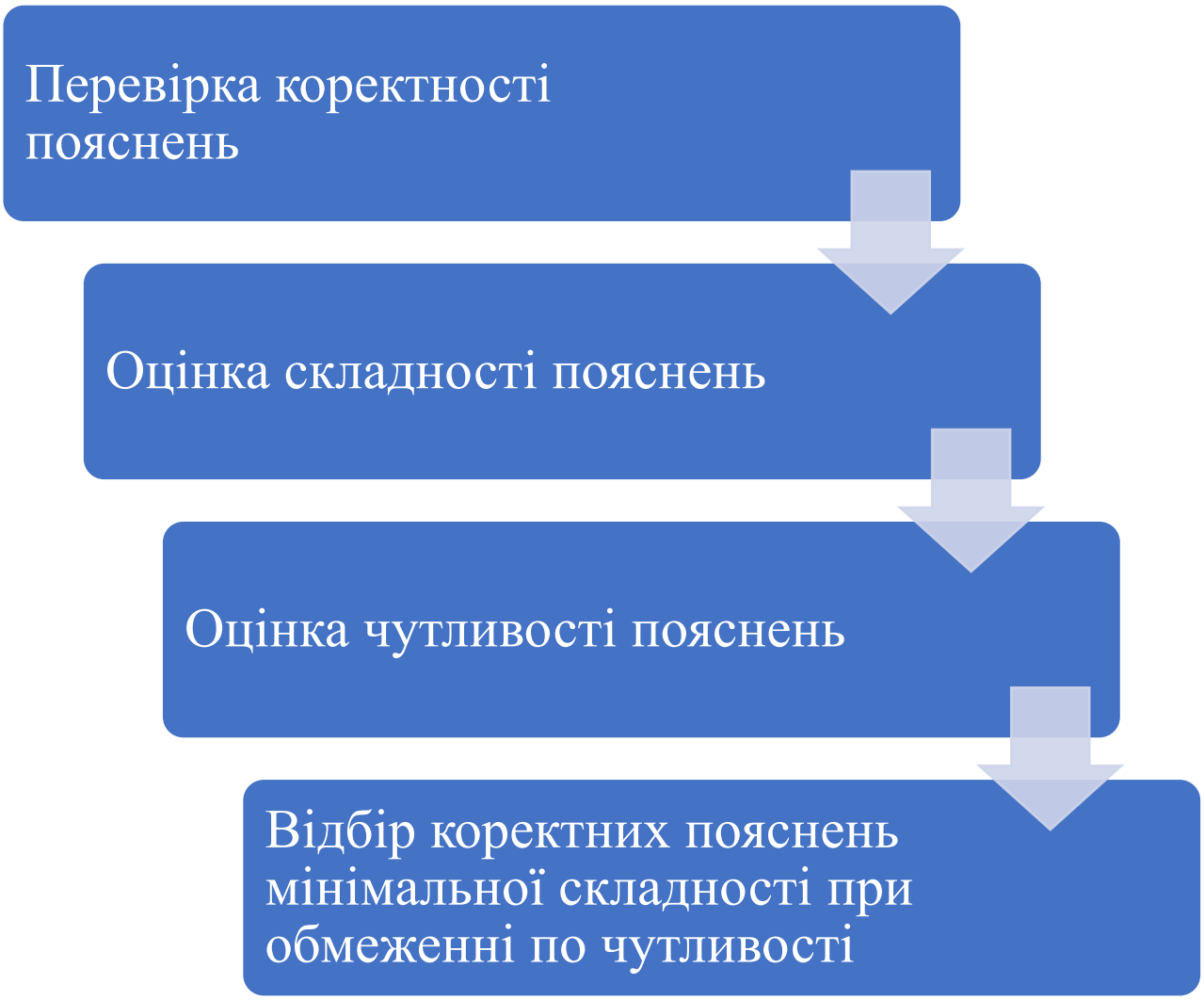

Рисунок 2.1 – Етапи удосконаленого можливісного методу оцінки пояснень

3 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОЦІНКИ ПОЯСНЕНЬ

3.1 Опис інформаційної технології

Пропонується використовувати наступну схему інформаційної технології для оцінки пояснень в інтелектуальній системі. Першочергово відбувається підготовка вхідних даних. Система отримує вхідні дані від користувача або інших джерел та проводить їх попередню обробку. На цьому етапі також відбувається перевірка та очищення даних від можливих артефактів. Після чого дані приводяться до формату, що зручний для подальшої обробки.

На основі коректних вхідних даних відбувається побудова пояснень. Система використовує різні алгоритми та моделі для аналізу даних і генерації пояснень. Результатом цього етапу є структуроване пояснення, яке включає ключові фактори, враховані моделлю.

Після чого проводиться оцінка згенерованих пояснень, використовуючи зокрема метод можливісної оцінки пояснень. Цей підхід дозволяє визначити важливість та вплив окремих чинників на результати моделі.

З отриманих пояснення відбираються найбільш прості пояснення для заданих обмежень по їх чутливості. Використовуючи метод відбору, визначаються фактори з найвищою чутливістю.

В результатів передбачається генерація звіту, який містить детальні пояснення та їхні оцінки. Створений звіт представляє користувачеві результати аналізу, включаючи згенеровані пояснення, результати оцінки та рейтинг чутливості факторів. Цей звіт призначений для ознайомлення користувача з висновками та обґрунтуванням прийнятих рішень моделі. Окрім цього, важливою частиною процесу є створення можливостей для взаємодії користувача з отриманими поясненнями з метою покращення їхнього розуміння та довіри до моделі.

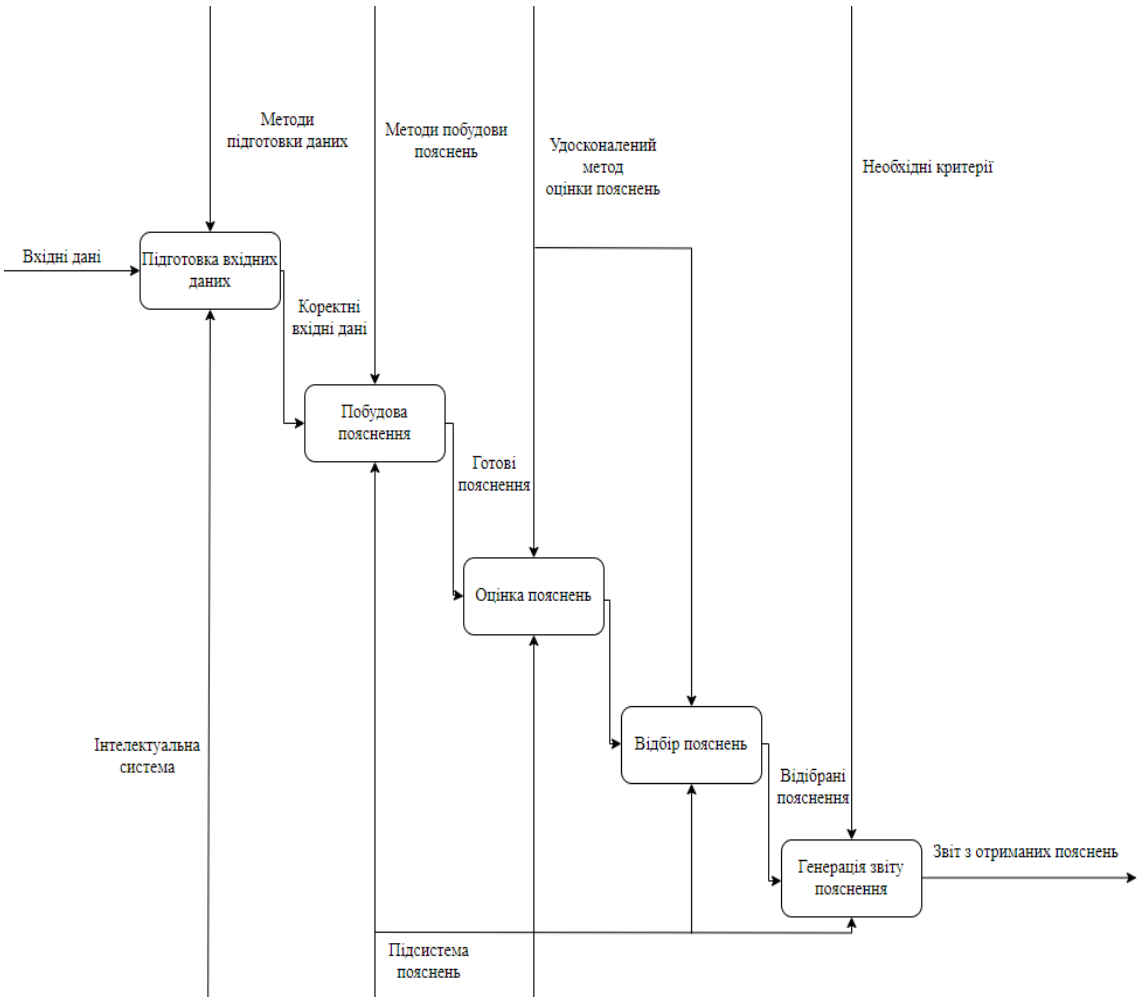

Рисунок 3.1 – Схема технології оцінки пояснень на основі критеріїв чутливості, коректності, складності

3.2 Імплементація інформаційної технології оцінки пояснень

В основі Docker лежить концепція контейнеризації, яка передбачає створення ізольованих середовищ в межах однієї операційної системи для окремих додатків. Це надає розробникам та системним адміністраторам безліч переваг, сприяючи гнучкості, портативності та ефективності використання ресурсів. Docker створює автономні контейнери для додатків. Ці контейнери об'єднують код програми, середовище виконання, системні бібліотеки та

налаштування, незалежно від основної хост-системи. Така ізоляція дає кілька ключових переваг.

Рисунок 3.2 – Переваги Docker

Окрім цих основних переваг, Docker надає багату екосистему інструментів та сервісів, які організовують життєвий цикл контейнерів. Docker Hub, центральний репозиторій, містить мільйони готових образів контейнерів, доступних для розгортання. Docker Compose спрощує створення додатків, що вимагають багато контейнерів визначаючи їхню конфігурацію в одному файлі. Docker Swarm та Kubernetes ще більше розширюють можливості

оркестрування контейнерів у масштабі, керуючи та координуючи розгортання контейнерів на кластерах машин.

Continuous Integration/Continuous Deployment (CI/CD) є важливими практиками в сучасній розробці програмного забезпечення, що дозволяють командам створювати високоякісне програмне забезпечення ефективніше та надійніше. Ці практики є особливо важливими у складних та масштабних проектах, де інтеграція та розгортання програмних компонентів є критично важливими для успіху проекту.

Continuous Integration (CI) – це практика частого злиття всіх локальних копій розробників в єдину систему з подальшим автоматизованим процесом збірки та тестування. Такий підхід дозволяє командам виявляти проблеми інтеграції на ранніх стадіях циклу розробки, забезпечуючи швидкий зворотній зв'язок про стан програмного забезпечення.

Ключові принципи CI включають підтримку єдиного репозиторію вихідного коду, автоматизацію процесів збірки та тестування, а також забезпечення перевірки кожного коміту автоматизованою збіркою. Завдяки частій інтеграції коду та запуску автоматизованих тестів, CI допомагає командам своєчасно виявляти та вирішувати проблеми, що призводить до покращення якості програмного забезпечення та пришвидшення циклів розробки.

Continuous Delivery/Deployment (CD) – є продовженням процесу CI, зосереджуючись на автоматизації доставки та розгортання релізних версій програмного забезпечення. Безперервна доставка забезпечує автоматизації процесу релізу, гарантуючи, що програмне забезпечення завжди знаходиться в стані готовності до розгортання. Це дозволяє командам випускати нові релізи програмного забезпечення передбачувано і стабільно, з гнучкістю приймати рішення про частоту випусків на основі бізнес-вимог. Безперервне розгортання просуває автоматизацію далі, забезпечуючи часте та автоматичне введення програмного забезпечення в експлуатацію.

Обидві практики CD спрямовані на скорочення часу та зусиль, необхідних для впровадження нових функцій та оновлень, зберігаючи при цьому високий рівень стабільності та надійності. Одним з найпоширеніших інструментів для реалізації CI/CD є Jenkins.

Jenkins – це популярний сервер автоматизації з відкритим вихідним кодом, який дозволяє розробникам безперервно створювати, тестувати та розгортати програмні проекти. Jenkins підтримує сотні плагінів, які інтегруються з різними інструментами та платформами, що робить його універсальним і потужним інструментом для безперервної інтеграції та доставки. Jenkins допомагає розробникам автоматизувати та впорядкувати життєвий цикл розробки програмного забезпечення, підвищити якість та надійність програмного забезпечення.

Саме тому для імплементації запропонованих рішень було вирішено використовувати перелічені інструменти.

Створення нової мережі Docker з назвою 'jenkins', яка використовується для ізольованої комунікації між контейнерами.

У команді docker run для створення контейнера 'jenkins-docker', використано кілька параметрів для досягнення необхідних цілей. Параметр --rm гарантує, що контейнер буде автоматично видалений після його завершення, що допомагає управлінню ресурсами. Опція --detach дозволяє запустити контейнер в фоновому режимі, забезпечуючи, що він не блокує термінал або інтерфейс командного рядка. Використання --privileged надає контейнеру розширені права, що є необхідним для функціонування Docker-in-Docker. Параметр --network jenkins підключає контейнер до створеної мережі 'jenkins', що забезпечує ізоляцію та можливість спілкування з іншими контейнерами в цій мережі. За допомогою --network-alias docker контейнер отримує аліас у мережі, що дозволяє легко звертатися до нього з інших контейнерів. Налаштування --env DOCKER_TLS_CERTDIR=/certs встановлює змінну середовища для шляху до TLS-сертифікатів Docker,

забезпечуючи безпечну комунікацію. Використання параметрів --volume створює томи для зберігання сертифікатів Docker і даних Jenkins, що дозволяє зберегти ці дані навіть після перезапуску або видалення контейнера. Нарешті, опція --publish 2376:2376 публікує порт контейнера на хості, дозволяючи зовнішньому доступу до сервісів, що запущені всередині контейнера.

```
docker network create jenkins

docker run --name jenkins-docker --rm --detach ^
  --privileged ^
  --network jenkins ^
  --network-alias docker ^
  --env DOCKER_TLS_CERTDIR=/certs ^
  --volume jenkins-docker-certs:/certs/client ^
  --volume jenkins-data:/var/jenkins_home ^
  --publish 2376:2376 ^
  docker:dind
```

Рисунок 3.3 – Команда для встановлення Docker image

Також в Dockerfile описано основні команди для виконання. Встановлення базового образу Jenkins, оновлення списку пакетів і встановлення необхідних пакети, таких як Python 3, pip, git, curl тощо. Встановлення Docker CLI, встановлення плагінів Jenkins, зокрема Blue Ocean та Docker Workflow.

```
FROM jenkins/jenkins:2.426.2-jdk17

USER root

RUN apt-get update && apt-get install -y \
    python3 \
    python3-pip \
    python3-venv \
    lsb-release \
    git \
    curl

RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc \
    https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
    signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
    https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list

RUN apt-get update && apt-get install -y docker-ce-cli

USER jenkins

RUN jenkins-plugin-cli --plugins "blueocean docker-workflow"

RUN mkdir -p /var/jenkins_home/my_custom_directory
```

Рисунок 3.4 – Dockerfile для встановлення контейнера

```yaml
version: '3'

services:
  jenkins-blueocean:
    image: myjenkins-blueocean:2.426.2-1
    container_name: jenkins-blueocean
    restart: on-failure
    networks:
      - jenkins
    environment:
      - DOCKER_HOST=tcp://docker:2376
      - DOCKER_CERT_PATH=/certs/client
      - DOCKER_TLS_VERIFY=1
    volumes:
      - jenkins-data:/var/jenkins_home
      - jenkins-docker-certs:/certs/client:ro
    ports:
      - "8080:8080"
      - "50000:50000"

networks:
  jenkins:
    external:
      name: jenkins

volumes:
  jenkins-data:
  jenkins-docker-certs:
```

Рисунок 3.5 – Docker Compose YAML файл

В основі ефективності Jenkins лежить Jenkinsfile, фундаментальний компонент конвеєрної функціональності Jenkins. Можна вважати, що Jenkinsfile – це детальна сценарій, що визначає весь процес створення застосунка або системи. Він забезпечує централізований і прозорий огляд конвеєра розгортання програмного забезпечення, охоплюючи всі етапи – від отримання актуального коду, останньої версії до розгортання доопрацьованого застосунку.

Завдяки ясності та доступності більше не потрібно шукати розрізнені скрипти або розшифровувати загадкові команди. Така прозорість не тільки спрощує спільну роботу, а й дає командам контроль над порядком завдань та інструментів, що використовуються в конвеєрі, забезпечуючи послідовність і передбачуваність кожної збірки.

Pipeline – це важливий компонент процесу CI/CD. Це набір автоматизованих кроків, яких дотримуються команди розробників програмного забезпечення для створення, тестування та розгортання програмних додатків. Pipeline – це стандартизований процес, який гарантує, що кожна зміна коду проходить через один і той же набір кроків, від розробки до випуску, забезпечуючи узгодженість і надійність.

Крім того, файл Jenkinsfile не статичний, а оновлюється разом із процесом розробки і підтримки програмного продукту. Кожна зміна відстежується і зберігається, що дає змогу повертатися до попередніх версій або безперешкодно працювати над покращеннями. Такий динамізм дає змогу Jenkinsfile відповідати ітеративній природі розробки програмного забезпечення, що розвивається.

```
pipeline {
    agent any
    stages {
        stage('Checkout') {
            steps {
                    git branch: 'main', url: 'https://github.com/
            }
        }

        stage('Build') {
            steps {
                script {
                    sh '''
                    python3 -m venv venv
                    .venv/bin/activate
                    python3 -m pip install -r requirements.txt
                    '''
                }
            }
        }

        stage('Test') {
            steps {
                sh '''
                .venv/bin/activate
                python3 main.py
                '''
            }
        }
```

Рисунок 3.6 – Фрагмент Jenkinsfile для автоматизації процесу розгортання удосконаленого методу пояснень

Використання вказаних інструментів та практик в контексті Jenkins та Docker є ключовим для забезпечення ефективного процесу.управління ресурсами, безпеки та спрощення процесу розгортання. Визначення та

виконання процедур через Dockerfile та Jenkinsfile оптимізує розгортання і тестування удосконаленого методу оцінки пояснень. Застосування Docker і Jenkins забезпечує однорідність середовища по всьому ланцюжку від розробки до розгортання. Це особливо важливо для імплементації удосконаленого методу оцінки пояснень, де консистентність в обробці та аналізі даних є ключовою. Оскільки інтелектуальні системи часто є складними та багатошаровими, використання Jenkins забезпечує чіткий контроль над кожним кроком процесу. Це важливо для забезпечення якості та надійності оцінки пояснень.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА МЕТОДУ ОЦІНКИ ПОЯСНЕНЬ

4.1 Програмна реалізація методу

Python – широко використовувана, високорівнева, універсальна, інтерпретована, динамічна мова програмування. Вона розроблена, щоб бути простою, читабельною та виразною, підтримуючи декілька парадигм програмування, таких як об'єктно-орієнтоване, імперативне, функціональне та процедурне. Python має велику і всеосяжну кількість стандартних бібліотек, а також багатий набір сторонніх бібліотек та фреймворків для різних областей, таких як веб-розробка, аналіз даних, наукові обчислення і машинне навчання.

PyTorch – популярний фреймворк машинного навчання з відкритим вихідним кодом, заснований на бібліотеці Torch. PyTorch надає дві основні функції: бібліотеку тензорних обчислень, яка підтримує прискорення графічного процесора та автоматичне диференціювання, та бібліотеку глибоких нейронних мереж, яка пропонує гнучкий та модульний API для побудови та навчання різних моделей. PyTorch також має багату екосистему інструментів і бібліотек, які розширюють його функціональність і підтримують розробку в галузі комп'ютерного зору, обробки природної мови, генеративного моделювання тощо.

PyTorch широко використовується для досліджень і розробок, оскільки дозволяє швидко створювати прототипи, динамічні графіки обчислень і легке налагодження. PyTorch також підтримує розгортання у виробництві, оскільки пропонує інструменти для серіалізації, оптимізації та обслуговування моделей. Також сумісний з різними хмарними платформами та сервісами машинного навчання, що забезпечує безперебійну розробку та легке масштабування.

Scikit-learn, або sklearn, є однією з найбільш популярних і широко використовуваних бібліотек машинного навчання в Python. Ця бібліотека

відрізняється великою кількістю алгоритмів, простотою використання та гнучкістю, що робить її ідеальним інструментом для розробників і аналітиків даних на всіх рівнях вмінь. Scikit-learn часто використовується в різноманітних областях, таких як фінанси, медецина, рекомендаційні системи. Його гнучкість та легкість у використанні роблять його ідеальним вибором для швидкої розробки прототипів та складних систем аналізу даних.

Саме тому було для програмної реалізації було вирішено використовувати описані технології. Вони є одними з провідних технологій у цій галузі, і продовжують розвиватися і вдосконалюватися, додаючи нові функції та можливості.

Деякі фрагменти коду програмної реалізації оцінки пояснень наведено на рисунках 4.1-4.7.

```
def stage_1(expl, num_items):
    for _ in range(num_items):
        is_correct = correctness_index(probability)
        explanations.append((expl, is_correct))
    return explanations



def stage_2(explanations):
    return [(item_id, correctness, complexity()) for item_id, correctness in explanations]
```

Рисунок 4.1 – Фрагмент коду програмної реалізації першого та другого етапів

```python
def stage_3(explanations):
    return [(item_id, correctness, expl_complexity, sensitivity())
            for item_id, correctness, expl_complexity in explanations]



def stage_4(explanations, sensitivity_threshold=0.1):
    mean_sensitivity = sum(e[3] for e in explanations) / len(explanations)
    lower_bound = mean_sensitivity * (1 - sensitivity_threshold)
    return sorted([e for e in explanations if e[1] and e[3] >= lower_bound], key=lambda x: x[2])
```

Рисунок 4.2 – Фрагмент коду програмної реалізації третього та четвертого етапів

```python
import pandas as pd
from surprise import Dataset, Reader, SVD, accuracy
from surprise.model_selection import train_test_split

data_path = 'data/input.csv'
df = pd.read_csv(data_path)
reader = Reader(rating_scale=(df['rating'].min(), df['rating'].max()))
data = Dataset.load_from_df(df[['userID', 'itemID', 'rating']], reader)
trainset, testset = train_test_split(data, test_size=0.25)
algo = SVD()
algo.fit(trainset)
predictions = algo.test(testset)
accuracy.rmse(predictions)
predicted_rating = algo.predict(user_id, item_id).est
```

Рисунок 4.3 – Фрагмент коду програмної реалізації завантаження даних

```python
def Explainer(method: str, model, dataset_tensor: torch.tensor, **params):
    default_params = {
        'lime': {'dataset_tensor': dataset_tensor, 'kernel_width': 0.75,
                 'std': float(np.sqrt(0.05)), 'mode': 'tabular',
                 'sample_around_instance': True, 'n_samples': 1000,
                 'discretize_continuous': False}
    }

    if method not in explainer_classes:
        raise NotImplementedError("This method has not been implemented, yet.")

    explainer_params = initialize_parameters(method, default_params, params.get(method))

    if method == 'lime':
        return explainer_classes[method](model.predict, **explainer_params)

    return explainer_classes[method](model, **explainer_params)
```

Рисунок 4.4 – Фрагмент коду програмної реалізації пояснення

```python
def initialize_parameters(method, default_params, user_params):
    params = default_params.get(method, {})
    if user_params:
        params.update(user_params)
    return params
```

Рисунок 4.5 – Фрагмент коду програмної реалізації ініціалізації параметрів

```python
class ExplanationEvaluator:


    def __init__(self, config: dict, input_data, target_labels, prediction_model, explanation_method):
        self.config = config
        self.input_data = input_data
        self.target_labels = target_labels
        self.prediction_model = prediction_model
        self.explanation_method = explanation_method
        self.predicted_label = self.config['predicted_label']
        self.ground_truth_importances = self._get_ground_truth_importances()
        self.explanation_features = self._compute_explanation()
```

Рисунок 4.6 – Фрагмент коду програмної реалізації оцінки пояснень

```
1 usage
def _get_ground_truth_importances(self):

    if hasattr(self.prediction_model, 'return_ground_truth_importance'):
        return self.prediction_model.return_ground_truth_importance(self.input_data).detach().numpy().reshape(1, -1)
    return None

1 usage
def _compute_explanation(self):

    explanation = self.explanation_method.get_explanation(self.input_data.float().reshape(1, -1),
                                                          label=self.predicted_label)
    return explanation.detach().numpy().flatten()
```

Рисунок 4.7 – Фрагмент коду програмної реалізації оцінки пояснень

4.2 Експериментальна перевірка методу

Відповідно до описаних технологій було виконано експериментальна перевірка удосконаленого методу оцінки пояснень.

Був налаштований Docker, що забезпечило надійне середовище для контейнеризації. В рамках Docker був запущений контейнер Jenkins, що забезпечує безперервну інтеграцію та платформу доставки для ефективної автоматизації завдань.

Запрограмований алгоритм на мові Python був успішно виконаний з використанням конвеєра Jenkins, продемонструвавши його функціональність і сумісність в рамках встановлених налаштувань.

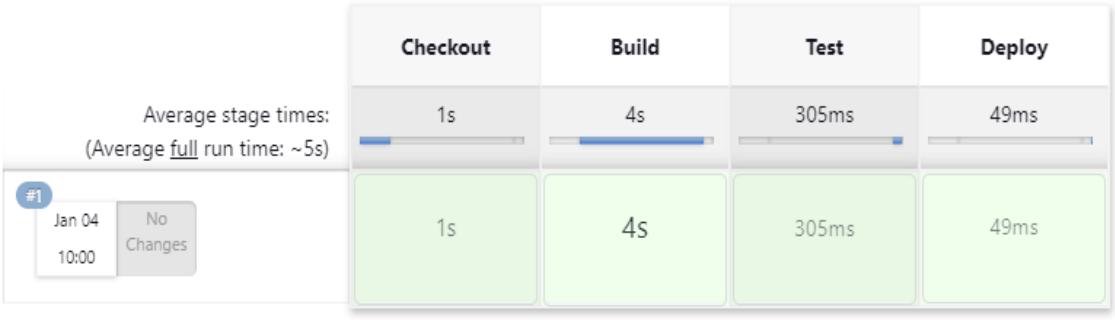

Рисунок 4.8 – Приклад візуалізації успішного розгортання удосконаленого методу пояснень

Відповідно до набору даних, який розглядався було скориговано та задано початкові параметри. Також варто мати на увазі, що в залежності від набору даних та подальшого отримання рішень можливо буде необхідним скоригувати модель для побудови рішення.

Отримані пояснення проходять перевірку на коректність.

```
explanations_stage_1 = stage_1(expl, num_items)
df_stage_1 = pd.DataFrame(explanations_stage_1, columns=['Item_id', 'Correctness'])
display(df_stage_1)
```

Рисунок 4.9 – Виконання першого етапу удосконаленого методу оцінки пояснень

	Item_id	Correctness
0	726	False
1	621	True
2	712	True
3	571	True
4	211	True
5	401	True
6	681	True
7	190	True
8	152	False
9	422	True
10	206	True
11	696	False
12	194	True
13	885	False
14	802	False
15	874	False
16	218	True
17	123	False
18	521	False
19	563	True

Рисунок 4.10 – Приклад результатів реалізації першого етапу удосконаленого методу оцінки пояснень

Відібрані коректні пояснення проходять оцінку складності.

```
explanations_stage_2 = stage_2(explanations_stage_1)
df_stage_2 = pd.DataFrame(explanations_stage_2, columns=['Item_id', 'Correctness', 'Complexity'])
display(df_stage_2)
```

Рисунок 4.11 – Виконання другого етапу удосконаленого методу оцінки пояснень

	Item_id	Correctness	Complexity
0	726	False	7
1	621	True	20
2	712	True	20
3	571	True	26
4	211	True	15
5	401	True	6
6	681	True	22
7	190	True	11
8	152	False	7
9	422	True	17
10	206	True	8
11	696	False	29
12	194	True	22
13	885	False	9
14	802	False	32
15	874	False	17
16	218	True	32
17	123	False	20
18	521	False	28
19	563	True	17

Рисунок 4.12 – Приклад результатів реалізації другого етапу удосконаленого методу оцінки пояснень

Відповідно до третього етапу удосконаленого алгоритму проводить оцінка чутливості.

```
explanations_stage_3 = stage_3(explanations_stage_2)
df_stage_3 = pd.DataFrame(explanations_stage_3, columns=['Item_id', 'Correctness', 'Complexity', 'Sensitivity'])
display(df_stage_3)
```

Рисунок 4.13 – Виконання третього етапу удосконаленого методу оцінки пояснень

	Item_id	Correctness	Complexity	Sensitivity
0	726	False	7	0.479763
1	621	True	20	0.400830
2	712	True	20	0.389605
3	571	True	26	0.400103
4	211	True	15	0.724117
5	401	True	6	0.504061
6	681	True	22	0.462924
7	190	True	11	0.735659
8	152	False	7	0.399755
9	422	True	17	0.632785
10	206	True	8	0.769283
11	696	False	29	0.678808
12	194	True	22	0.454510
13	885	False	9	0.559381
14	802	False	32	0.657771
15	874	False	17	0.682620
16	218	True	32	0.378688
17	123	False	20	0.455478
18	521	False	28	0.621968
19	563	True	17	0.477658

Рисунок 4.14 – Приклад результатів реалізації третього етапу удосконаленого методу оцінки пояснень

Всі наявні пояснення, що пройшли оцінку коректності, складності та чутливості проходять відбір по максимальному відхиленні чутливості n (0.1) і сортуються за складність від менш складних до складніших, відповідно.

```
sensitivity_threshold = 0.1
explanations_stage_4 = stage_4(explanations_stage_3, sensitivity_threshold)
df_stage_4 = pd.DataFrame(explanations_stage_4, columns=['Item_id', 'Correctness', 'Complexity', 'Sensitivity'])
display(df_stage_4)
```

Рисунок 4.15 – Виконання четвертого етапу удосконаленого методу оцінки пояснень

	Item_id	Correctness	Complexity	Sensitivity
0	401	True	6	0.504061
1	206	True	8	0.769283
2	190	True	11	0.735659
3	211	True	15	0.724117
4	422	True	17	0.632785

Рисунок 4.16 – Приклад результатів реалізації четвертого етапу удосконаленого методу оцінки пояснень

Результати, отримані вході експериментальної перевірки, показують, що кожен з пояснень має свої особливості щодо складності та чутливості.

ID 401 має середню складність (6) та помірну чутливість (0.504061), що може вказувати на його збалансованість у використанні та стабільність у реагуванні на зовнішні фактори.

ID 206, зі складністю 8 та чутливістю 0.769283, може свідчити про необхідність додаткових зусиль для його використання, але також про його високу адаптивність до змінних умов.

ID 190 та 211, маючи високу складність (відповідно 11 та 15) та схожу чутливість (0.735659 та 0.724117), можуть бути підходящими для складних сценаріїв, де потрібна детальна розробка та глибокий аналіз.

ID 422, з найвищою складністю (17) у цьому списку, але з помірною чутливістю (0.632785), може вимагати особливих знань та умінь для ефективного використання, але не надто чутливий до зовнішніх змін.

ВИСНОВКИ

Основним результатом роботи стало удосконалення методу оцінки пояснень. Удосконалений метод зосереджується на відборі найбільш релевантних та зрозумілих пояснень, що характеризуються мінімальною складністю і враховують необхідний рівень чутливості. Такий метод сприяє підвищенню прозорості та доступності інформації в ІІС.

Було розроблено і описано інформаційну технологію для впровадження удосконаленого методу. Була надана схема технології та перелік необхідних інструментів, що дозволяють застосовувати розроблений метод в практичних сценаріях.

Була виконана програмна реалізація удосконаленого методу оцінки пояснень і проведено експериментальну перевірку. Це дозволило не лише продемонструвати практичну застосовність розробленого методу, але й оцінити його ефективність та надійність у реальних умовах.

Отримані результати можуть, бути використанні в майбутніх дослідженнях. Результати роботи можуть бути застосовані для розробки нових інтелектуальних інформаційних систем, а також для удосконалення існуючих систем з метою підвищення їх здатності до надання зрозумілих та коректних пояснень користувачам.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до передатестаційної практики для студентів усіх форм навчання спеціальності 122 – Комп’ютерні науки, освітньо-професійної програми "Інформаційні управляючі системи та технології" / Упоряд.: Чалий С.Ф., Євланов М. В., Чала О. В. - Харків: ХНУРЕ, 2021

2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки: Структура та правила оформлювання:. - К.: Держстандарт України, 2015. - 31с.

3. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складаня. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.

2. Чалий С. Ф. Моделювання пояснень щодо рекомендованого переліку об’єктів з урахуванням темпорального аспекту вибору користувача. Чалий С. Ф., Лещинський В. О., Лещинська І. О. Системи управління, навігації та зв’язку. 2019. Т. 6. № 58. С. 97-101.

3. Чалий С. Ф. Концепція формування пояснень в рекомендаційних системах за принципом білого ящику. Чалий С. Ф., Лещинський В. О., Лещинська І. О. Системи Управління, Навігації Та Зв’язку. Збірник Наукових Праць, 3(55), 156–160.

4. Чалий С. Ф., Лещинський В. О. Метод можливісного оцінювання пояснення в системі штучного інтелекту. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. 2023. № 2. С. 95– 101.

5. Чалий С. Ф., Лещинський В. О. Оцінка чутливості пояснень в інтелектуальній інформаційній системі. Системи управління, навігації та зв’язку. Збірник наукових праць.2023. №2. С. 165–169.

6. Чалий С. Ф., Лещинський В. О., Лещинська І. О. (2020). Модель пояснення в інтелектуальній інформаційній систе-мі на основі концепції узгодженості знань. Вісник Національного технічного університету "ХПІ". Сер. : Системний аналіз, управління та інформаційні технології, 1 (3), 19-23.

7. Чалий С.Ф., Лещинський В.О., Лещинська І.О. Декларативно-темпоральний підхід до побудови пояснень в інтелектуальних інформаційних системах. Вісник Нац. техн. ун-ту "ХПІ": зб. наук. пр. Темат. вип.Системний аналіз, управління та інформаційні технології. Харків: НТУ «ХПІ». 2020. № 2(4). С.51–56.

8. Adadi, A., and Berrada, M. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access 6 (2018).

9. Alonso J. M., Castiello C., Mencar C. A Bibliometric Analysis of the Explainable Artificial Intelligence Research Field. In: Medina, J., et al. Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations. IPMU. Communications in Computer and Information Science. 2018. Vol 853. P. 3–15.

10. Arrieta, A.B., Dìaz- Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S, Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. Information Fusion, 58, pp.82-115, doi.org/10.1016/j.inffus.2019.12.012

11. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and MÃžller, K.-R. How to explain individual classification decisions. Journal of Machine Learning Research, 11 (Jun):1803–1831, 2010.

12. Chalyi S., Leshchynskyi V., Leshchynska I. Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system. EUREKA: Physics and Engineering. 2019. Vol. 4. P. 34–40.

13. Chalyi S, Leshchynskyi V. Probabilistic counterfactual causal model for a single input variable in explainability task. Advanced Information Systems. 2022. №7(3), P. 54–59. DOI: https://doi.org/10.20998/2522-9052.2023.3.08.

14. Chalyi S., Leshchynskyi V. Possible evaluation of the correctness of explanations to the end user in an artificial intelligence system. A.I.S.2023. №7. P. 75–79

15. Dabkowski, P. and Gal, Y. Real time image saliency for black box classifiers. In Advances in Neural Information Processing Systems, pp. 6967–6976, 2017.

16. Engelbrecht Andries P. Computational Intelligence: An Introduction. NJ: John Wiley & Sons, 2007. 632 p

17. Gunning D., Aha D. DARPA's Explainable Artificial Intelligence (XAI) Program. AI Magazine. 2019. Vol.40 (2). P. 44-58.

18. Markus, A. F., Kors, J. A., and Rijnbeek, P. R. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. Journal of Biomedical Informatics 113 (2021).

19. Miller, T., Howe, P., and Sonenberg, L. Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences. In IJCAI 2017 Workshop on Explainable Artificial Intelligence (2017).

20. Pörner, N., Schütze, H., and Roth, B. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In ACL (2018).

21. Rong, Y., Leemann, T., Borisov, V., Kasneci, G., and Kasneci, E. A consistent and efficient evaluation strategy for attribution methods. In ICML (2022), vol. 162, PMLR.

22. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. IEEE Transactions on Neural Networks and Learning Systems 28, 11 (2017)

23. Sokol, K., and Flach, P. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In FAccT (2020), ACM.

24. Tintarev, N., and Masthoff, J. Explaining recommendations: Design and evaluation. In Recommender Systems Handbook. Springer, 2015.

25. Topin, N., and Veloso, M. Generation of policy-level explanations for reinforcement learning. In AAAI (2019).

26. Verma, M., and Ganguly, D. LIRME: locally interpretable ranking model explanation. In SIGIR (2019), ACM.