

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук

## КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему Забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю

Галузь знань 12 – Інформаційні технології

Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки

Шифр і назва спеціальності

Виконав: студент 2 курсу, група КНм-20-1



Підпис

О.В. Дудар

Ініціали, прізвище

Керівник: к.ф-м.н., доцент кафедри КН



Підпис

В.Ц. Михалевський

Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН



Підпис

Р.О. Багрій

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

\_\_\_\_\_ 2021 р.



Підпис

О.В. Бармак

Ініціали, прізвище

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук  
Освітній ступінь магістр  
Галузь знань 12 – Інформаційні технології  
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
комп'ютерних наук

(підпис)

д.т.н., професор О.В. Бармак

«»2021 року

### ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ МАГІСТРА

1. Тема дипломної роботи магістра: «Забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю»
2. Завдання видано студенту Дударю Олександрові Віталійовичу  
(прізвище, ім'я, по батькові)
3. Керівник роботи к.фіз.-мат.н., доцент Міхалевський Віталій Цезарійович  
(прізвище, ім'я, по батькові)
4. Затверджені наказом університету від «25»серпня2021р. № 102
5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – теоретично обґрунтувати, розробити та апробувати систему для забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю. Для досягнення мети роботи треба провести дослідження існуючих підходів підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю.

## Реферат

Кваліфікаційна робота магістра присвячена розробці інформаційної системи для забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю.

**Актуальність дослідження** зумовлена стрімким розвитком використання засобів автоматизації, таких як електронні системи обробки заявок в тій чи іншій тематиці, автоматизацією усіх процесів життєдіяльності людини.

Процес пошуку виконавця для виконання заявки на прибирання відходів в громадських місцях займає багато часу і потребує немало зусиль. Автоматизація даного процесу звільняє велику кількість часу і дозволяє підвищити кількість заявок, що позитивно вплине на основну діяльність, збереження екологічної рівноваги у сфері переробки відходів.

Тому система автоматизації забезпечення екологічної рівноваги у сфері переробки відходів є не тільки рекомендаційною до впровадження, а й абсолютною необхідністю, що суттєво оптимізує процес збору та переробки відходів.

Але існуючи на даний час системи автоматизації забезпечення екологічної рівноваги у сфері переробки відходів не використовують оптимізаційної моделі забезпечення підтримки екологічної рівноваги.

Цю проблему можна розв'язати за рахунок розробки оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Для розв'язання визначеної проблеми нами буде розроблена оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Мета і задачі роботи** полягає у розробці оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Для досягнення поставленої мети визначені наступні задачі дослідження:

- побудова моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів;
- для дослідження практичної ефективності розробити інформаційну модель системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Об'єктом дослідження** є процес збору, представлення, обробки, зберігання, передачі та доступу до інформації системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Предметом дослідження** є оптимізаційна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Методи дослідження** застосовані для вирішення поставлених завдань: для реалізації оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів. – методи збору, представлення, обробки, зберігання, передачі та доступу до інформації в комп'ютерних системах.

### **Наукова новизна одержаних результатів.**

В результаті проведеної роботи були отримані такі результати:

- набули подальшого розвитку існуючі моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів;
- досліджено практичну ефективність моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Практичне значення одержаних результатів. На основі розробленої моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів для підтвердження результатів дослідження була створена інформаційна система забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Спираючись на отримані результати дослідження на базі створеної інформаційної системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів, отримано наступні результати:

- розроблена модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів продемонструвала ефективність за результатами дослідження;

- обрана оптимальна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Апробація результатів кваліфікаційної роботи** За темою кваліфікаційної роботи магістра автором виконана *наукова публікація*: Дудар О.В. Інформаційна система для забезпечення підтримки екологічної рівноваги / Дудар О.В., Міхалевський В. Ц., Скрипник Т.К. // Збірник наукових праць за матеріалами XIII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». – Хмельницький, 2021. – С. 321-323.[39].

**Структура та обсяг роботи.** Дипломна робота магістра складається з завдання, реферату, змісту, вступу, 4 розділів, висновків, переліку посилань із 39 найменування та додатків. Загальний обсяг дипломної роботи магістра становить 99 сторінок, з них 77 сторінок основного тексту та 21 сторінка додатків. В роботі наведено 15 рисунків та 1 таблиця.

**Ключові слова:** методи, оптимізація, відходи, переробка, екологія.

## Зміст

<b>Перелік скорочень</b> .....	4
Вступ.....	5
Розділ 1 Аналіз сучасного стану проблеми підтримки екологічної рівноваги у сфері переробки відходів.....	8
1.1 Аналіз предметної області.....	8
1.2 Дослідження інформаційного забезпечення підтримки екологічної рівноваги.....	12
1.3 Дослідження математичних моделей для автоматизації процесу підтримки екологічної рівноваги.....	23
1.4 Постановка задачі.....	39
Висновки до розділу 1.....	39
Розділ 2 Розробка інформаційної технології для забезпечення підтримки екологічної рівноваги.....	40
2.1 Створення моделі оптимізації переробки відходів.....	40
2.2 Побудова моделі підтримки екологічної рівноваги.....	41
2.3 Аналіз вимог до інформаційної моделі системи з підтримки екологічної рівноваги у сфері переробки відходів.....	51
Висновки до розділу 2.....	54
Розділ 3 Проектування реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.....	55
3.1 Основні вимоги до проектування інформаційної системи.....	55
3.2 Проектування загальної схеми інформаційної системи.....	55
3.3 Проектування інтерфейсної частини системи.....	57
Висновки до розділу 3.....	60

Розділ 4 Дослідження ефективності інформаційної технологій для забезпечення підтримки екологічної рівноваги .....	62
4.1 Обґрунтування мови програмування для розробки реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.....	62
4.2 Обґрунтування вибору програмної технології для розробки реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж .....	66
4.3 Розробка системи прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж .....	69
4.4 Розробка модуля нейронної мережі системи прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж .....	72
Висновки до розділу 4.....	73
Загальні висновки.....	75
Перелік використаних джерел .....	<b>Error! Bookmark not defined.</b>
Додатки	

## Перелік скорочень

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
WWW	World Wide Web
CSS	Каскадні таблиці стилів
KPM	Кваліфікаційна робота магістра
HTML	Гіпертекстова мова розмітки
JSON	JavaScript Object Notation
ООП	Об'єктно-орієнтоване програмування

## Вступ

Кваліфікаційна робота магістра присвячена розробці інформаційної системи для забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю.

**Актуальність дослідження** зумовлена стрімким розвитком використання засобів автоматизації, таких як електронні системи обробки заявок в тій чи іншій тематиці, автоматизацією усіх процесів життєдіяльності людини.

Процес пошуку виконавця для виконання заявки на прибирання відходів в громадських місцях займає багато часу і потребує немало зусиль. Автоматизація даного процесу звільняє велику кількість часу і дозволяє підвищити кількість заявок, що позитивно вплине на основну діяльність, збереження екологічної рівноваги у сфері переробки відходів.

Тому система автоматизації забезпечення екологічної рівноваги у сфері переробки відходів є не тільки рекомендаційною до впровадження, а й абсолютною необхідністю, що суттєво оптимізує процес збору та переробки відходів.

Але існуючи на даний час системи автоматизації забезпечення екологічної рівноваги у сфері переробки відходів не використовують оптимізаційної моделі забезпечення підтримки екологічної рівноваги.

Цю проблему можна розв'язати за рахунок розробки оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Для розв'язання визначеної проблеми нами буде розроблена оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Мета і задачі роботи** полягає у розробці оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Для досягнення поставленої мети визначені наступні задачі дослідження:

- побудова моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів;
- для дослідження практичної ефективності розробити інформаційну модель системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Об'єктом дослідження** є процес збору, представлення, обробки, зберігання, передачі та доступу до інформації системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Предметом дослідження** є оптимізаційна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Методи дослідження** застосовані для вирішення поставлених завдань: для реалізації оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів. – методи збору, представлення, обробки, зберігання, передачі та доступу до інформації в комп'ютерних системах.

#### **Наукова новизна одержаних результатів.**

В результаті проведеної роботи були отримані такі результати:

- набули подальшого розвитку існуючі моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів;
- досліджено практичну ефективність моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Практичне значення одержаних результатів. На основі розробленої моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів для підтвердження результатів дослідження була створена інформаційна система забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Спираючись на отримані результати дослідження на базі створеної інформаційної системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів, отримано наступні результати:

- розроблена модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів продемонструвала ефективність за результатами дослідження;
- обрана оптимальна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Апробація результатів кваліфікаційної роботи** За темою кваліфікаційної роботи магістра автором виконана *наукова публікація*: Дудар О.В. Інформаційна система для забезпечення підтримки екологічної рівноваги / Дудар О.В., Міхалевський В. Ц., Скрипник Т.К. // Збірник наукових праць за матеріалами XIII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». – Хмельницький, 2021. – С. 321-323.[39].

**Структура та обсяг роботи.** Дипломна робота магістра складається з завдання, реферату, змісту, вступу, 4 розділів, висновків, переліку посилань із 39 найменування та додатків. Загальний обсяг дипломної роботи магістра становить 99 сторінок, з них 77 сторінок основного тексту та 21 сторінка додатків. В роботі наведено 15 рисунків та 1 таблиць.

## **Розділ 1**

### **Аналіз сучасного стану проблеми підтримки екологічної рівноваги у сфері переробки відходів**

#### **1.1 Аналіз предметної області**

Переробка – це процес перетворення відходів у нові матеріали та нові засоби. Ця концепція зазвичай передбачає відновлення енергії з відходів. Обробка матеріалу залежить від його здатності відновлювати характеристики свого початкового стану. [1] Це альтернатива «традиційній» обробці відходів, яка економить матеріали та зменшує викиди парникових газів. Це також може запобігти розтраті потенційно корисних матеріалів та зменшити споживання свіжої сировини за рахунок зменшення споживання енергії, забруднення повітря (від спалювання) та забруднення води (від звалищ).

Переробка є ключовим компонентом сучасного скорочення відходів і третім компонентом системи класифікації відходів «зменшення, повторне використання та переробка» [2, 3]. Він сприяє екологічній стійкості, вилучаючи сировину та перенаправляючи викиди відходів в економічну систему [4]. Існують деякі стандарти ISO щодо переробки, наприклад ISO 15270:2008 для пластикових відходів і ISO 14001:2015 щодо контролю середовища переробки.

Матеріали, що підлягають переробці, включають багато видів скла, паперу, картону, металу, пластику, шин, текстилю, акумуляторів та електроніки. Повторне використання компосту та інших біологічно розкладних відходів (таких як харчові та садові відходи) також є формою переробки [5]. Перероблені матеріали або відправляються в центри переробки, або виймаються з контейнерів біля узбіччя, потім сортуються, очищаються та переробляються на нові матеріали для виробництва нових продуктів.

В ідеалі переробка може забезпечити нові запаси тих самих матеріалів — наприклад, використаний офісний папір буде перетворено на новий офісний папір, а використаний пінополістирол буде перетворений на новий

полістирол. Певні типи матеріалів, наприклад металеві банки, можна відновлювати знову і знову на невизначений термін без втрати чистоти [6]. Для інших матеріалів це зазвичай складно або занадто дорого (у порівнянні з виробництвом того самого продукту із сировини чи інших джерел), тому «переробка» багатьох продуктів і матеріалів передбачає їх повторне використання у виробництві різних матеріалів (наприклад, картон). Іншим способом переробки є відновлення складових матеріалів із складних продуктів, оскільки вони мають внутрішню цінність (наприклад, свинець в автомобільних акумуляторах і золото в друкованих платах) або їх небезпечні властивості (наприклад, видалення та повторне використання з термометрів і термостатів) (HG).

#### Постачання

Для того, щоб програма переробки запрацювала, необхідний великий і стабільний запас вторинних матеріалів. Для створення таких матеріалів використовувалися три законодавчі варіанти: обов'язковий збір вторинної сировини, законодавство про контейнерне зберігання та заборона сміття. Закони про обов'язковий збір встановлюють цілі щодо переробки для міст, як правило, таким чином, що певний відсоток матеріалів має бути переданий із міського потоку відходів до цільової дати. Місто зобов'язане наполегливо працювати для досягнення цієї мети [5].

Законодавство про депозитну тару передбачає повернення певної тари (як правило, скляної, пластикової, металеві). При купівлі товарів у такій тарі додається невелика надбавка, і споживачі можуть повернути тару, повернувшись до пункту збору. Цим програмам вдалося створити середній рівень відновлення 80% [34]. Незважаючи на такі хороші результати, передача витрат на збирання від органів місцевого самоврядування до галузей і споживачів викликала рішучу опозицію в певних сферах [5] – наприклад, виробники несуть відповідальність за переробку своєї продукції. У Європейському Союзі директива WEEE вимагає від виробників побутової електроніки відшкодовувати витрати на переробників [35].

Інший спосіб збільшити постачання вторинної сировини – заборонити утилізацію деяких матеріалів як відходів, як правило, включаючи відпрацьоване масло, старі акумулятори, шини та садові відходи. Це може створити життєздатну економіку для належної утилізації. Переконайтеся, що існує достатня кількість послуг з утилізації, щоб відповідати вимогам, інакше такі заборони можуть призвести до збільшення кількості незаконних звалищ [5].

#### Урядове замовлення

Для збільшення та підтримки попиту на вторинну сировину також використовувалися чотири форми законодавства: мінімальна потужність для вмісту вторинної сировини, специфікації використання, політика закупівель та маркування продуктів переробки [5].

Як мінімальні вимоги щодо вмісту вторинної сировини, так і коефіцієнтів використання збільшили попит, що змушує виробників включати переробку у свою діяльність. Вимоги до змісту передбачають, що певний відсоток нових продуктів повинен складатися з перероблених матеріалів. Коефіцієнт використання є більш гнучким варіантом: галузь може досягти своїх цілей утилізації на будь-якому етапі своєї діяльності і навіть може підписувати контракти на переробку в обмін на продаж кредиту. Противники цих методів посилалися на їх значне збільшення вимог до звітності і стверджували, що вони позбавляють індустрію гнучкості [5, 36].

Уряд використовує власну купівельну спроможність, щоб збільшити попит на переробку за допомогою «політики закупівель». Ці політики є або «фіксованою стороною», яка резервує певні витрати на оброблені продукти, або планом «цінової переваги», який передбачає більший бюджет при купівлі вторсировини. Інші нормативні акти можуть бути спрямовані на конкретні ситуації: наприклад, у Сполучених Штатах Агентство з охорони навколишнього середовища наказало закуповувати нафту, папір, шини та будівельні ізоляційні матеріали з перероблених або відновлених джерел, коли це можливо [5].

Остаточним державним регулюванням, що збільшує попит, є маркування продуктів переробки. Коли виробникам потрібно вказати кількість перероблених

матеріалів (включаючи упаковку), які вони містять на своїй упаковці, споживачі можуть зробити більш усвідомлений вибір. Споживачі з достатньою купівельною спроможністю можуть вибрати більш екологічні варіанти та заохочувати виробників до збільшення кількості вторинної сировини у своїй продукції для збільшення попиту. Якщо стандартизована етикетка про переробку визначає, як і де продукт переробляється, це також може мати позитивний вплив на переробку[5].

#### Якість переробленого сміття

Якість переробленого сміття є одним із головних викликів успіху довгострокового бачення зеленої економіки та реалізації нульових відходів. Взагалі кажучи, це стосується того, скільки його складу складається з цільових матеріалів у порівнянні з нецільовими матеріалами та іншими матеріалами, які не підлягають переробці [36]. Сталь та інші метали переробляються вищої якості; за оцінками, дві третини нової сталі отримують з переробленої сталі. Ймовірно, що перероблятимуться лише цільові матеріали, тому більше нецільових матеріалів і матеріалів, які не підлягають переробці, можуть зменшити кількість переробленої продукції [36]. Великій частині нецільових матеріалів і матеріалів, які не підлягають вторинній переробці, важко досягти «високої якості» вторинної переробки; якщо якість вторинної сировини погана, це, швидше за все, завершиться наступний цикл або в більш екстремальних умовах кейсів, відправляються на іншу переробку відбору або звалища [36]. Наприклад, щоб сприяти відновленню прозорого скла, існують суворі обмеження на кольорове скло, яке потрапляє в процес переплавлення. Іншим прикладом є переробка пластику. Такі продукти, як пластикова харчова упаковка, зазвичай переробляються на продукти нижчої якості, а не в ту саму пластикову харчову упаковку.

Якість вторинної сировини не тільки сприяє високоякісній переробці, але також може принести значні переваги для довкілля за рахунок скорочення, повторного використання та збереження продуктів зі звалищ [36]. Якісна переробка може підтримати економічне зростання, максимізуючи витрати на

відходи [36]. Вищий рівень доходу від продажу високоякісної вторинної сировини може принести величезну цінність місцевим органам влади, домогосподарствам і підприємствам [36]. Прагнення до якісної переробки може також підвищити впевненість споживачів і бізнесу в інвестиціях та управлінні відходами та ресурсами.

У ланцюжку поставок переробки багато дій, і кожна дія вплине на якість переробки [21]. Виробники відходів, які розміщують нецільові та непереробні відходи на переробних станціях, можуть вплинути на якість кінцевого потоку переробки та потребувати додаткових зусиль для переробки цих матеріалів на пізніх стадіях процесу переробки [21]. Різні системи збирання призведуть до різного рівня забруднення. При об'єднанні кількох матеріалів потрібні додаткові зусилля, щоб класифікувати їх на різні потоки, що може значно знизити якість кінцевого продукту [21]. Транспортування та ущільнення матеріалів також можуть ускладнити це. Незважаючи на вдосконалення технології та якості переробки, сортувальна установка все ще не на 100% ефективна в розділенні матеріалів [21]. Коли матеріал намокає під час зберігання на відкритому повітрі, це також може спричинити проблеми для переробників. Можуть знадобитися додаткові етапи сортування, щоб задовільно зменшити кількість нецільових і непридатних для вторинної переробки матеріалів [21].

## **1.2 Дослідження інформаційного забезпечення підтримки екологічної рівноваги**

Проведемо огляд та аналіз предметної області, розглянувши флагманські програмні рішення для забезпечення підтримки екологічної рівноваги.

***Програмне забезпечення ТОВ "Фірма Інтеграл" СЕРІЇ "ЕКОЛОГ"***  
Фірма «Інтеграл» - лідер у галузі розробки програмних засобів у галузі охорони навколишнього середовища. Серед них програми щодо розрахунків, для ведення баз даних, для оформлення документів за затвердженими формами, довідкові програми, все це в галузі охорони атмосферного повітря, безпечного поводження

з відходами, оцінки забруднення водних об'єктів, санітарної акустики, санітарно-гігієнічного моніторингу.

На сьогоднішній день програми для екологів, розроблені Фірмою «Інтеграл», вирішують весь спектр завдань у галузі промислової екології, що піддаються автоматизації. Серед них програми щодо розрахунків, для ведення баз даних, для оформлення документів за затвердженими формами, довідкові програми, все це в галузі охорони атмосферного повітря, безпечного поводження з відходами, оцінки забруднення водних об'єктів, санітарної акустики, санітарно-гігієнічного моніторингу. Програми застосовуються для розробки проектної природоохоронної документації: проекти нормативів ПДВ, ПНООЛР, ПДВ, розділ проекту «Перелік заходів з охорони навколишнього середовища», проект СЗЗ підприємства тощо; під час заповнення форм статистичної звітності підприємства, при розрахунку та оформленні плати за негативний вплив на довкілля. Користувачі програм – екологи промислових підприємств, розробники природоохоронної та проектної документації, експерти державних природоохоронних відомств, санітарні лікарі, викладачі екологічних дисциплін у вишах. Програми мають усі необхідні погодження та сертифікати.

Серед наших розробок є унікальні на ринку продукти. Серед них:

- уніфікована програма розрахунку забруднення атмосфера (УПРЗА) «Еколог» для розрахунку максимальних концентрацій забруднюючих речовин в атмосфері (розрахунку розсіювання), в т.ч. з урахуванням впливу забудови;
- розрахунковий блок «Середні» для розрахунку середніх за тривалий період концентрацій забруднюючих речовин в атмосфері;
- розрахунковий блок «Ризики» для оцінки ризиків для здоров'я людини за фактором забруднення атмосфери;
- система «Еколог-Місто» для проведення зведених розрахунків забруднення атмосфери в масштабах міста (регіону);
- програмний комплекс «Еколог-Шум» для розрахунку шуму від промислових джерел та транспорту.

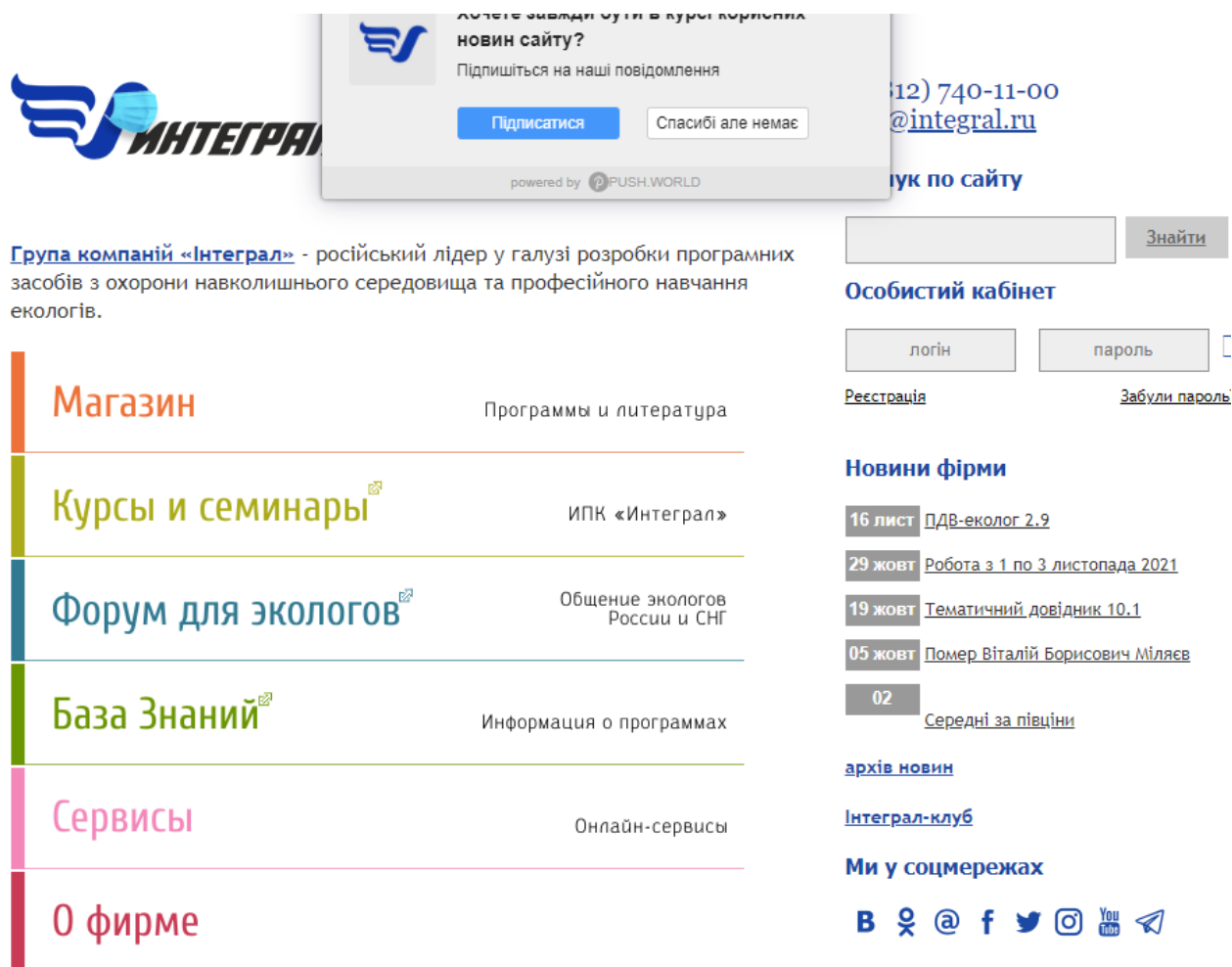


Рисунок 1.1 – Програмне забезпечення ТОВ "Фірма Інтеграл" СЕРПІ "ЕКОЛОГ" [23].

*EcoReport* – це сучасні високотехнологічні програми для екологів. «Просто про складне» - базовий принцип побудови екологічних програм проекту, що забезпечує наочність представлення даних та зручність виконання розрахунків.

Екологічне програмне забезпечення націлене на максимальну підтримку користувача та повністю автоматизує процес виконання розрахунків та формування екологічної звітності. Вхідні дані піддаються алгоритмізації обчислень, програми для екологів контролює дані, що вводяться користувачем, запобігаючи можливості введення апріорі некоректних значень, інформуючи і звертаючи увагу людини на необхідність виконання тих чи інших дій. Суворя

відповідність екологічному законодавству, точність реалізації розрахункових методик, повнота довідкової інформації та якісна підтримка забезпечують успішний розвиток екологічного проекту EcoReport. Екологічне програмне забезпечення EcoReport (Екорепорт) забезпечує розрахунок плати за негативний вплив на довкілля. EcoReport - це програма для екологів, яка допоможе виконати розрахунок екологічних платежів та сформувати звіт "Розрахунок плати за негативний вплив на навколишнє середовище".

Програми для екологів: розрахунок плати за негативний вплив на довкілля. Екологічні платежі **EcoReport** **розрахунок** **НВОС** Екологічне програмне забезпечення EcoReport розрахунок НВОС призначений для розрахунку плати за негативний вплив на навколишнє середовище. Програма для екологів реалізує розрахунок екологічних платежів та формування звіту "Розрахунок плати за негативний вплив на навколишнє середовище" у «паперовому» (вивантаження у MS Excel) та «електронному» вигляді (XML файли). Програма відповідає вимогам наказу Укртехнагляду від 27.03.2008 р. N 182.

Екологічна програма представлена у 4 версіях та розрахована як на професійних екологів, так і на співробітників, які поєднують виконання розрахунку плати за НВОС зі своїми основними обов'язками. Існує безкоштовна версія програми.

Програми для екологів: звітність за формою 2-ТП (відходи) **EcoReport 2-ТП (відходи)**. Програма для екологів EcoReport 2ТП (відходи) призначена для створення державної статистичної звітності за формою 2-ТП (відходи). Екологічна програма забезпечує як «ручне» формування звітів шляхом вибору забруднюючих речовин із довідника ФККО та введення річних обсягів їх утворення, так і автоматизоване – шляхом імпорту даних із квартальних звітів «Розрахунок плати за НВОС». Перед покупкою завантажуюмо та дивимося безкоштовну версію програми.

Програми для екологів: звітність за формою 2-ТП (повітря) **EcoReport 2-ТП (повітря)** Програма для екологів EcoReport 2-ТП (повітря)

призначена для створення державної статистичної звітності за формами 2-ТП (повітря) та 2-ТП-повітря (Термінова). Програма забезпечує як «ручне» формування звітів, так і автоматизоване шляхом імпорту даних із квартальних електронних звітів «Розрахунок плати за НВОС». Перед покупкою можна спробувати безкоштовну версію програми.

Програмне забезпечення для екологів розрахункові модулі ***EcoReport модуль Лівнівка***. Безкоштовна програма для розрахунку зливових стоків. Програма призначена для розрахунку мас утворення забруднюючих речовин при неорганізованому скиданні у водні об'єкти для всіх типів територій та підприємств.

Програми для екологів: електронна звітність ***EcoReport Електронна звітність***. Система EcoReport Електронна звітність забезпечує можливість подання звітів в Управління Укрприроднагляду по Сумській області з телекомунікаційних каналів зв'язку. Система реалізує можливість надання звітності (розрахунок плати за НВОС, 2-ТП та ін.) як у повністю електронному форматі, з підписанням документа ЕЦП, так і у суміщеному вигляді, коли звіт здається в електронному вигляді без ЕЦП з паралельним надсиланням паперового звіту за допомогою звичайної пошти. Єдина безкоштовна повнофункціональна система подання екологічної звітності в електронному вигляді.

- Питання експерту
- Інструкція з оформлення заявки на ПС
- Документи
- Сертифікати
- Знижки
- Заміна версій
- Демо-версії
- Корисні статті
- Форум
- Фотографії
- Задати питання
- Еко-посилання

**Для зв'язку із Зубовою Наталією Ростиславівною використовувати Skype [zoubova](#)**

**Дякую за розуміння.**

**Ми** пропонуємо комплексні рішення та готові програмні засоби в екології та природокористуванні , призначені:

- для автоматизації робіт відділів охорони навколишнього середовища промислових об'єднань/підприємств;
- для проведення оцінки стану навколишнього середовища при прийнятті управлінських та проектних рішень та здійснення контролю за їх виконанням на рівні міста/регіону;
- для розробки проектів ПДВ , ПДС та НООЛР ;
- для організації навчальних процесів у ВНЗ, на курсах підвищення кваліфікації та професійної перепідготовки кадрів промислових підприємств та організацій;
- для інформаційної підтримки в галузі охорони навколишнього середовища та природокористування .

**З** спеціалістами ЗАТ НВП "ЛОГУС" готові надати кваліфіковану допомогу Вашій організації при проведенні наступних робіт:

- розрахунок суми платежів за негативний вплив на довкілля
- ведення журналів ПІД - 1, 2,3 та ПІД - 11,12,13 і т.д.
- складання звітів 2-тп (повітря) , 2-тп (відходи) , 2-тп (водгосп)
- розробка проектів нормативів гранично допустимих викидів ( ПДВ )
- розрахунок класу небезпеки відходів
- розробка проектів нормативів освіти відходів та лімітів на їх розміщення ( НООЛР )
- підготовка Технічного звіту про незмінність виробничого процесу, використовуваної сировини та поводження з відходами
- розробка проектів нормативно-допустимих скидів ( ПДВ )
- розробка проектів санітарно-захисних зон ( СЗЗ ) підприємств та організацій

**Пошук**

**часті питання**

**Драйвер ключа HASP у Windows XP SP2**

Рисунок 1.2 – EcoReport [24].

*Hydromantis Environmental Software Solutions, Inc.* надає досвідчені послуги моделювання для галузей промисловості обробки води та стічних вод. Hydromantis Environmental Software Solutions, Inc. забезпечує спеціалізовані моделювання послуг для води та водопостачання виробничих індустрій. Ми повинні наші вдосконалені моделювання освітлення з розсудливими випробуваннями і в-глибокий пізнання. Вони практичні в нашому виконанні, конкретні в наших зауваженнях, і тому в нашій execution. Наші експертизи можуть бути використані для water treatment plant design, commissioning, operational performance, and planning improvements.

*Програмне забезпечення:*

### **GPS-X**

Premium Water & Wastewater Modelling та Simulation Software GPS-XTM був першим комерційно випущеним динамічним симулятором очисної установки стічних вод. GPS-XTM був першим комерційно виконаним динамічним wastewater treatment plant simulator.

***SimuWorks***

Operator Simulators для Advanced Training and Plant Analysis SimuWorks™ представляє еволюцію у використанні моделювання обробки води та стічних вод та моделювання технології, розширюючи її використання поза технічними офісами. SimuWorks™ є промисловим першим - на флеш-симулятор для води і водопостачання термічного планів. Розвиваючи технологію Hydromantis 'GPS-X™, промисловість є найбільш математичним моделюванням і simulation software, SimuWorks™ вивчає і структуру аналізу можливостей до нових сили. SimuWorks™ являють собою еволюцію у використанні води.

***Toxchem***

Predict & Report Toxic Emissions Планування, проектування та оптимізація схем процесу зменшити викид в атмосферу. Планування, дизайн і оптимізація процесів schemes to reduce air emission.



## News

Рисунок 1.3 – Hydromantis Environmental Software Solutions, Inc [25].

*Склад пакету ТЕРПООV*

### **RTI**

програма для розрахунку втрат тепла та інфільтрації приміщеннями будівель(тепловтрат), енергопаспорт будівліПризначення та область застосування:

Програма призначена:

Для визначення втрат тепла будинками та спорудами різного призначення з урахуванням втрат тепла на інфільтрацію з метою складання теплового балансу

на опалювальні прилади під час проектування систем опалення. При цьому можливо:

- виконати розрахунок опору теплопередачі багат шарових конструкцій, що захищають;
- вибрати товщину ізоляції стін;
- перевірити температуру на межах шарів;
- визначити "точку роси" на внутрішніх поверхнях та у кутах приміщень.

Для підготовки "Енергетичного паспорта будівлі":

- має можливість виконання розрахунку параметрів енергетичного паспорта будівель та споруд;
- формує звіт з Енергопаспорту згідно з "додатком Д", СП 50.13330.2012 та СНіПом 23-02-2012;
- редагований шаблон пояснювальної записки з подальшим експортом до MS Word, Excel, OpenOffice (13 різних форматів).

**VSV** – програма для аеродинамічного розрахунку систем вентиляції, аспірації та пневмотранспорту.

Програма призначена для аеродинамічного розрахунку систем вентиляції, аспірації та пневмотранспорту. Для складання завдання (підготовки вихідних даних) потрібна наявність схем проєктованих вентиляційних систем із зазначенням довжин ділянок та витрат повітря на кінцевих ділянках. Вхідними даними для розрахунку є опис систем вентиляції та вимоги до неї (швидкість у магістральних повітроводах та у відгалуженнях). Системи вентиляції - припливні, витяжні з (жорсткими повітропроводами) круглими (гнучкими, круглими/овальними, стислими) або прямокутними повітроводами. Системи аспірації та пневмотранспорту – витяжні з круглими повітроводами. Існує можливість поділу ділянки системи з постійною витратою на кілька розрахункових без використання фіктивного трійника.

**VIBROS** – програма для розрахунку викидів котельної труби. Програма призначена для автоматизованої підготовки даних по котельних установках (при розробці розділу проєкту «Охорона навколишнього

середовища») для подальшого виконання розрахунків концентрацій шкідливих викидів в атмосфері за уніфікованими програмами розрахунку забруднення атмосфери (УПРЗА), погоджених із МГО ім. А.І. Воєйкова. Програма надає можливість застосуванням диктуючих параметрів різних пристроїв котельні, промодельовати різні проектні варіанти роботи котельні, проаналізувати отримані результати моделювання котельні, що проектується або працює, з метою вибору запланованого (оптимального) режиму роботи котельної установки. У випадку діючої котельні на основі та/або інструментальних вимірів та конструктивних даних, виконати моделювання, прийняти рішення, як, змінюючи режими роботи окремих пристроїв котельні, можливо мінімізувати викиди шкідливих речовин в атмосферу (екологія).

**STOL** – програма для розрахунку повітрообміну підприємств громадського харчування. Призначення та сфера застосування Програма призначена для визначення параметрів припливного повітря, складання повітряного балансу в основному для гарячого цеху підприємств громадського харчування (їдалень, кафе, ресторанів тощо) з урахуванням встановленого модульованого та/або іншого обладнання в гарячому цеху, на лінії роздачі. Може поставлятися як у складі інших програм комплексу ТЕРПООV (ТЕПЛООВ), так і окремо від програм комплексу ТЕРПООV (ТЕПЛООВ).

**BOLER** – програма для теплових розрахунків бойлерних установок. Призначення та сфера застосування. Програма призначена для виконання теплових розрахунків бойлерних установок, що складаються з швидкісних водоводяних односекційних теплообмінників, пароводяних, двоходових та чотирьох ходових ПП1 та ПП2. Може поставлятися як у складі інших програм комплексу ТЕРПООV (ТЕПЛООВ), так і окремо від програм комплексу ТЕРПООV (ТЕПЛООВ).

**KALOR** – програма для розрахунку калориферів та повітронагрівачів, підбіртипових припливних камер. Призначення та сфера застосування Програма призначена для підбору індивідуальних калориферних установок, що забезпечують підігрів заданої кількості повітря на необхідний перепад

температур для: секцій підігріву припливних камер; повітряно-теплових завіс; пропарювальних камер. Програма має накопичувач кліматичних даних місця забудови, допоміжну інформацію щодо параметрів повітря залежно від кліматичних даних. Компонування установки виконується програмно з набору запропонованих компоновок, або конструюється користувачем.

ПОТОК  
РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
Microsoft Partner

ГОЛОВНА    ДЕМО ВЕРСІЯ    ЦІНИ    НОВИНИ    ФОРУМ

Инженерные расчёты для умных инженеров

Можем научить !!!

Ласкаво просимо!!!

[" - пакет програм для розрахунку систем опалення, вентиляції та теплових розрахунків.](#)

[ртифікат відповідності в системі ДЕРЖСТАНДАРТ Р № СС RA.RU.AB86.H01001.](#)

[дписка на повідомлення про оновлення.](#)

редовище використання MS Windows VISTA, Windows7, ndows8, Windows10

Мова сайту > english

логін:

пароль:

Реєстрація  
Нагадати пароль?

Склад пакету ТЕРПООV

ПОТОК

програма для розрахунку систем опалення, охолодження тепlopостачання калориферів та обладнання

останнє оновлення 26.11.2021

Новини  
Наші останні новини

21-10-2021, 13:46  
**Зміни у програмі Потік**  
>читати повністю

Рисунок 1.4– Пакет ТЕРПООV[26].

### 1.3 Дослідження математичних моделей для автоматизації процесу підтримки екологічної рівноваги

Найпростіший спосіб досягти поставленої мети роботи, затасувати для розв'язку поставлених завдань нейронну мережу, щоб спростити процес вибору методу переробки відходів і зменшити кількість помилок при прийнятті даного рішення.

Оскільки обраний для реалізації метод передбачає розробку нейронних мереж, концепцію нейронних мереж слід розглядати як частину загальної предметної області.

Штучна нейронна мережа (ANN), яку часто називають нейронною мережею (NN), — це обчислювальна система, натхненна біологічними нейронними мережами, які утворюють мозок тварин. [1]

Штучні нейронні мережі засновані на наборі пов'язаних одиниць або вузлів, які називаються штучними нейронами, які можуть вільно моделювати нейрони біологічного мозку. Кожна сполука, як синапс у біологічному мозку, може передавати сигнали іншим нейронам. Штучний нейрон, який отримує сигнал, потім обробляє його і може посилати сигнал підключеному до нього нейрону. «Сигналом» у з'єднанні є дійсне число, а вихід кожного нейрона обчислюється певною нелінійною функцією суми його входів. З'єднання називаються ребрами. Нейрони та ребра зазвичай мають ваги, які регулюються під час процесу навчання. Під час підключення вага збільшує або зменшує силу сигналу. Нейрони можуть мати поріг, тобто вони будуть посилати сигнал лише тоді, коли сукупний сигнал перевищує поріг. Зазвичай нейрони об'єднуються в шари. Різні шари можуть виконувати різні перетворення на своїх входах. Сигнал переміщується від першого шару (вхідного шару) до останнього шару (вихідного шару), можливо, після кількох раундів шарів.

Уоррен Маккалок і Уолтер Піттс (2) (1943) почали цю тему, створивши обчислювальні моделі для нейронних мереж. [3] Наприкінці 1940-х років Д. О. Хебб [4] запропонував гіпотезу навчання, засновану на механізмі

нейропластичності, а саме навчання Хебба. Фарлі та Уеслі А. Кларк [5] (1954) вперше використали комп'ютери (пізніше названі «калькуляторами») для моделювання мережі Хеббі. Розенблат [6] (1958) створив персептрон. [7] Першу багатoshарову функціональну мережу опублікували Івахненко і Лапа в 1965 році як груповий метод обробки даних. [8] [9] [10] Основні знання про безперервне зворотне поширення [8] [11] [12] [13] містяться в контексті теорії керування Келлі [14] у 1960 році та Брайсона [15] у 1961 році. принцип динамічного програмування.

У 1970 році Сеппо Лінненмаа опублікував загальний метод автоматичного диференціювання (AD) дискретних зв'язаних мереж вкладених диференціальних функцій [16] [17]. У 1973 році Дрейфус використав зворотне поширення, щоб налаштувати параметри контролера пропорційно градієнту помилки. [18] Алгоритм зворотного поширення Вербоса (1975) дозволяє практичне навчання багаторівневим мережам. У 1982 році він застосував метод Лінненмаа AD до нейронних мереж, який широко використовувався [11] [19]. Після Мінського та Пеперта (1969) [20] дослідження припинилися, і вони виявили, що базові персептрони не можуть впоратися з монополіями чи схемами, а комп'ютери не мають можливості обробляти корисні нейронні мережі.

Широкомасштабна інтеграція металооксидних напівпровідників (MOS), розроблена у вигляді додаткової технології MOS (CMOS), збільшила кількість МОП-транзисторів у цифровій електроніці. Це забезпечило більшу обчислювальну потужність для розробки практичних штучних нейронних мереж у 1980-х роках [21]

У 1992 році було введено максимальне об'єднання, щоб мінімізувати зсув інваріантності та допуск до деформації, тим самим допомагаючи розпізнаванню тривимірних об'єктів [22] [23] [24]. Шмідхубер прийняв багаторівневу мережеву ієрархію (1992 р.), Підготуйтеся заздалегідь на одному рівні за допомогою неконтрольованого навчання і вдосконалюйтеся шляхом перерозподілу.

Джеффри Хінтон (2006) запропонували використовувати безперервні двійкові або реальні приховані змінні шари та кінцеву машину Больцмана [26] для моделювання кожного шару для вивчення високорівневих уявлень. У 2012 році Нг і Дін створили мережу, яка вчиться розпізнавати більш просунуті поняття, такі як кішки, дивлячись на зображення без міток. Неконтрольована попередня підготовка та збільшення обчислювальної потужності від графічних процесорів і розподілених обчислень призвели до використання більших мереж, особливо для розпізнавання зображень і візуального, що називається «глибоким навчанням» [28].

Ciresan та його колеги (2010) [29] показали, що, незважаючи на проблему зникаючих градієнтів, графічні процесори дозволяють поширювати багат шарові нейронні мережі прямого зв'язку назад [30]. З 2009 по 2012 рік Академія наук почала займати призові місця в конкурсі Академії наук, наближаючись до людського рівня в різних завданнях, спочатку в розпізнаванні образів і машинному навчанні [31] [32]. Наприклад, двонаправлена та багатовимірна довготривала пам'ять (LSTM) Грейвса та інших [33] [34] [35] [36]. У 2009 році він переміг у трьох суміжних конкурсах на розпізнавання рукописного тексту і не володів трьома мовами, які раніше вивчалися [35] [34]. Ciresan та його колеги створили перший розпізнавач шаблонів, щоб досягти конкурентоспроможності людини/надлюдини в еталонах, таких як розпізнавання дорожніх знаків (IJCNN 2012) [37].

Нейронні мережі навчаються, обробляючи приклади. Кожен приклад містить відомі «вхідні дані» та «результати», які утворюють між ними зважену за ймовірністю асоціацію. Ці асоціації зберігаються в структурі даних самої мережі. Навчання нейронної мережі з даного прикладу зазвичай здійснюється шляхом визначення різниці між обробленим результатом мережі (зазвичай очікуваним) і цільовим результатом. Це помилка. Потім мережа коригує свої зважені асоціації відповідно до правил навчання і використовуючи це значення помилки. Постійні коригування змусять нейронну мережу давати результати, які все більше і більше схожі на цільові результати. Після внесення достатньої

кількості цих коригувань навчання може бути припинено за певними критеріями. Це називається контрольованим навчанням.

Така система «вчиться» виконувати завдання, розглядаючи приклади, зазвичай без написання певних правил для конкретних завдань.

Наприклад, під час розпізнавання зображень вони можуть навчитися розпізнавати зображення, які містять котів, аналізуючи приклади зображень, які вручну позначені як «кішки» або «без котів», і використовуючи результати, щоб ідентифікувати котів на інших зображеннях. Вони роблять це, не знаючи котів заздалегідь, наприклад, у них є шерсть, хвости, вуса і котячі мордочки. Замість цього вони автоматично створюватимуть ідентифікаційні ознаки з оброблених прикладів.

Штучні нейронні мережі складаються з штучних нейронів, які концептуально походять від біологічних нейронів. Кожен штучний нейрон має вхід і виробляє один вихід, який можна надіслати кільком іншим нейронам. Вхідними даними можуть бути значення ознак зовнішніх зразків даних, таких як зображення чи документи, або вихідні дані інших нейронів. Остаточний вихід нейронної мережі Вихід нейрона виконує такі завдання, як розпізнавання об'єктів на зображенні. Щоб знайти вихід нейрону, спочатку беремо зважену суму всіх входів, зважену вагами зв'язків від входів до нейрона. До цієї суми додаємо термін упередженості. Цю зважену суму іноді називають активацією. Потім ця зважена сума передається через (як правило, нелінійну) функцію активації для отримання вихідних даних. Початкові дані - це зовнішні дані, такі як зображення та документи. Кінцеві результати виконують завдання, наприклад, розпізнавання об'єкта на зображенні [40].

Штучний інтелект (ШІ) – це інтелект, який демонструють машини, а не природний інтелект тварин, включаючи людей. Провідні підручники зі штучного інтелекту визначають цю сферу як вивчення «інтелектуальних агентів»: будь-якої системи, яка сприймає навколишнє середовище та вживає заходів, щоб максимізувати свої цілі. Деякі популярні звіти використовують термін «штучний інтелект» для опису машин, які на них працюють. Наслідування «когнітивних»

функцій людей, пов'язаних з людським мисленням, таких як «навчання» та «вирішення проблем», але це визначення відкидається основними дослідниками штучного інтелекту [6].

До програм штучного інтелекту входять розширені веб-пошукові системи (наприклад, Google), системи рекомендацій (які використовуються YouTube, Amazon і Netflix), розуміння людської мови (наприклад, Siri та Alexa) та самокеровані автомобілі (наприклад, Tesla), Найвищий рівень автоматизованого прийняття рішень і змагання в стратегічних ігрових системах (таких як шахи) [2]. Оскільки машини стають потужнішими, завдання, які вимагають «інтелекту», зазвичай вилучаються з визначення штучного інтелекту, що є відомим явищем як ефект ШІ [3]. Наприклад, оптичне розпізнавання символів зазвичай виключається зі штучного інтелекту[4] і стало звичайною технологією [5].

Штучний інтелект був заснований як дисципліна в 1956 році, і в останні роки він пережив кілька хвиль оптимізму [6] [7], за якими послідували розчарування та втрата коштів (так звана «AI Winter») [8]. [9]. Про нові методи, успіхи та оновлене фінансування [7] [10]. З моменту свого заснування дослідження штучного інтелекту перевіряли багато різних методів, включаючи моделювання мозку, моделювання вирішення людських проблем, формальну логіку, великі бази даних та моделювання поведінки тварин.

У перші десятиліття 21-го століття високоматематичне статистичне машинне навчання домінує в цій галузі, і ця методика виявилася дуже успішною, допомагаючи вирішити багато складних проблем у промисловості та наукових сферах [11][10].

Різні підрозділи досліджень ШІ зосереджені на конкретних цілях і використанні конкретних інструментів. Традиційні цілі досліджень штучного інтелекту включають міркування, представлення знань, планування, навчання, обробку природної мови, сприйняття та здатність переміщати об'єкти та маніпулювати ними. - Пряма ціль. [12] Для вирішення цих проблем дослідники штучного інтелекту прийняли та інтегрували широкий спектр методів вирішення

проблем, включаючи пошук і математичну оптимізацію, формальну логіку, штучні нейронні мережі та методи, засновані на статистиці, ймовірності та економіці. ШІ також використовує інформатику, психологію, лінгвістику, філософію та багато інших галузей.

Це поле ґрунтується на припущенні, що людський інтелект «можна описати настільки точно, що він може моделювати машини.» [D] Це викликало філософську дискусію щодо ідей та етики створення штучного інтелекту за допомогою людського інтелекту. Ці питання обговорюються міфами, романами та філософією з давніх часів [14]. Наукова фантастика та футурологія також показують, що завдяки своєму величезному потенціалу та потужності штучний інтелект може стати загрозою людському існуванню [15] [16].

У машинному навчанні перцептрон — це алгоритм контрольованого навчання бінарних класифікаторів. Двійковий класифікатор — це функція, яка може визначити, чи належить вхід, представлений цифровим вектором, до певної категорії [1]. Це лінійний класифікатор, тобто алгоритм класифікації, який прогнозує на основі функції лінійного предиктора, що поєднує набір вагових показників з векторами ознак.

Алгоритм перцептрона був винайдений Френком Розенблатом [3] в Корнельській аеронавігаційній лабораторії в 1958 році за фінансової підтримки Управління військово-морських досліджень США [4].

Perceptron має бути машиною, а не програмою. Хоча його перша реалізація була реалізована в програмному забезпеченні IBM 704, пізніше вона була реалізована в спеціально розробленому апаратному забезпеченні, такому як Mark 1 Perceptron. Ця машина призначена для розпізнавання зображень: вона має масив із 400 фотоелементів, випадковим чином підключених до «нейронів». Вага кодується в потенціометрі і оновлюється за допомогою мотора під час тренування.

На прес-конференції, організованій ВМС США в 1958 році, Розенблат виступив із заявою про перцептрони, що викликало запеклі суперечки в молодому співтоваристві ШІ. За словами Розенבלата, New York Times

повідомила, що перцептрони це «зародок електронного комп'ютера, яка, як очікується, зможе сама ходити, говорити, бачити, писати і копіювати себе, усвідомлювати своє існування».

Хоча спочатку перцептрон здавався багатообіцяючим, незабаром він не міг навчити перцептрон розпізнавати багато типів шаблонів. В результаті дослідження в області нейронних мереж зупинилися на багато років, поки люди не зрозуміли, що нейронні мережі, які безпосередньо спілкуються з двома або більше шарами (також звані багат шаровими перцептронами), кращі за одношарові перцептрони (також звані багат шаровими перцептронами). Sensor) має більш потужні можливості обробки (також відомий як багат шаровий перцептрон). Називається одношаровим. Кульковий датчик).

Одношарові перцептрони можуть вивчати лише лінійно розділені моди. Щоб задати класифікацію з певною функцією активації кроку, вузол матиме лінію, що розділяє точки даних, що утворюють шаблон. Більше вузлів може створити більше розділових ліній, але ці лінії потрібно якимось чином об'єднати, щоб сформувати більш складну класифікацію. Другого шару перцептронів і навіть лінійних вузлів достатньо для вирішення багатьох інших нерозривних завдань.

У 1969 році відома книга під назвою «Перцептрони», написана в співавторстві Марвіна Мінського і Сеймура Пеппера, показала, що неможливо вивчити функції XOR таких типів мереж. Люди зазвичай (помилково) вважають, що вони також припускають, що багат шарова перцептронна мережа дасть подібні результати. Однак це не так, тому що і Мінськ, і Пеппер вже знають, що багат шарові перцептрони можуть створювати функції XOR. Проте, часто змішаний текст Мінськ/Паперт призвів до значного зниження інтересу та фінансування досліджень нейронних мереж. Минуло ще десять років, перш ніж дослідження нейронних мереж відродилися в 1980-х роках. Ця стаття була передрукована як «Perceptrons-Expanded Edition» у 1987 році, показуючи та виправляючи деякі помилки в оригінальному тексті.

Айзерман і його співавтори вже запропонували ядерний персептронний алгоритм у 1964 році [5]. Забезпечує граничні гарантії для алгоритму персептрона в загальних ситуаціях інтеграції, вперше надані Freund and Shapiro (1998) [1], а нещодавно Mohri і Rostamizade (2013), які розширили попередні результати та ввели нову межу L1 [6].

Персептрони — це спрощені моделі біологічних нейронів. Хоча повне розуміння поведінки нейронів зазвичай вимагає складності біологічних моделей нейронів, дослідження показали, що лінійні моделі, подібні до персептронів, можуть призвести до деяких видів поведінки, які спостерігаються у реальних нейронів [7].

Алгоритм кишенькового храповика (Gallant, 1990) вирішує проблему стабільності навчання персептрону і зберігає найкраще рішення «у вашій кишені». Тоді кишеньковий алгоритм повертає рішення в кишеню замість останнього рішення. Його також можна використовувати для неподільних наборів даних, де метою є знайти персептрони з невеликою кількістю помилок. Однак ці рішення видаються суто випадковими, тому кишеньковий алгоритм не буде поступово наближатися до них під час процесу навчання, і немає гарантії, що вони з'являться в межах заданої кількості кроків навчання.

Алгоритм Максовера (Wendemuth, 1995) є «надійним», оскільки він відповідає незалежно від (раніше) знання лінійного поділу набору даних [12]. У разі лінійного поділу це дозволить вирішити проблему навчання — якщо потрібно, навіть із найкращою стабільністю (максимальний запас між класами). Для неподільних наборів даних він поверне рішення з кількома помилками. У всіх випадках алгоритм поступово наближається до розв'язання в процесі навчання, він не запам'ятовує попередній стан і не стрибає випадково. Збіжність — це глобальна оптимальність розділеного набору даних і локальна оптимальність нерозривного набору даних.

Персептрон для голосування (Freund and Schapire, 1999) — це варіант, який використовує кілька зважених персептронів. Кожен раз, коли вибірка неправильно класифікується, алгоритм запускає новий персептрон і ініціалізує

ваговий вектор остаточною вагою останнього персептрона. Кожному персептрону також буде надано різну вагу, що відповідає тому, скільки прикладів вони правильно класифікували до неправильної класифікації, а кінцевим результатом буде зважене голосування для всіх персептронів.

В окремому завданні навчання персептрону також може мати на меті знайти найбільший роздільник між класами. Так званий персептрон оптимальної стабільності можна визначити за допомогою ітераційного навчання та схем оптимізації, таких як алгоритм Min-Over (Krauth and Mezard, 1987) [11] або AdaTron (Anlauf and Biehl, 1989). 13] AdaTron використовує той факт, що відповідна задача квадратичної оптимізації є опуклою. Персептрон оптимальної стабільності разом із трюком ядра є концептуальними основами машини опорного вектора.

Альфа-персептрон також використовує шар попередньої обробки з фіксованими випадковими вагами з пороговими вихідними одиницями. Це дозволяє персептрону класифікувати аналогові моделі, проектуючи їх у двійковий простір. Фактично, для достатньо великого простору проекції візерунки можуть стати лінійно розділеними.

Інший спосіб розв'язувати нелінійні задачі без використання кількох шарів — використовувати мережі вищого порядку (одиниці сигма-пі). У цьому типі мережі кожен елемент у вхідному векторі розширюється кожною парною комбінацією (другого порядку) мультиплікативного входу. Це можна поширити на мережі  $n$  порядку.

Однак майте на увазі, що найкращим класифікатором не обов'язково є той, який ідеально класифікує всі навчальні дані. Насправді, якщо ми маємо дані про обмеження еквівалентного гауссового розподілу раніше, то лінійне поділ у вхідному просторі є найкращим, і нелінійне рішення буде переоснащено.

Інші алгоритми лінійної класифікації включають Winnow, опорні векторні машини та логістичну регресію.

Штучний нейрон - це своєрідна математична функція, яка задумана як модель біологічного нейрона, тобто нейронної мережі. Штучний нейрон є

основною одиницею штучної нейронної мережі [1]. Штучні нейрони отримують один або кілька вхідних даних (що представляють збуджуючі постсинаптичні потенціали та гальмівні постсинаптичні потенціали в нейронних дендритах) і агрегують їх, щоб отримати вихід (або представляють нейрони, які передають по своїх аксонах) активацію потенціалів дії). Як правило, кожен вхід окремо зважується, і кількість передається за допомогою нелінійної функції, яка називається функцією активації або функцією передачі. Форма передатної функції зазвичай сигмоподібна, але вони також можуть мати форму інших нелінійних функцій, кусково-лінійних функцій або ступінчастих функцій. Вони також мають тенденцію бути монотонними, безперервними, диференційованими та обмеженими. Останнім часом також вивчаються немонотонні, необмежені й коливальні функції активації з багатьма нулями, які у багатьох задачах перевершують сигмоїдальні та ReLU-подібні функції активації [23]. Функція коливальної активації може покращити градієнтний потік в нейронній мережі і дозволяє вивчати функції з меншою кількістю нейронів. Наприклад, один нейрон GCU можна використовувати для вивчення функцій XOR з біполярним кодуванням [2]. Порогова функція надихнула на створення логічних клапанів, які називаються пороговою логікою; вона підходить для побудови логічних схем, подібних до обробки мозку. Наприклад, останнім часом для розвитку такої логіки широко використовуються нові пристрої, такі як мемристори [4].

Передатну функцію штучного нейрона не слід плутати з передатною функцією лінійної системи.

Першим штучним нейроном був пороговий логічний блок (TLU) або лінійний пороговий блок [10], вперше запропонований Уорреном Маккалоком і Волтером Піттсом у 1943 році. Ця модель спеціально розроблена як обчислювальна модель «нейронної мережі» в мозку. [11] Як передатну функцію він використав поріг, еквівалентний ступінчастій функції Хевісайда. Спочатку розглядалися лише прості моделі з двійковими входами та виходами, деякими обмеженнями щодо можливих ваг і більш гнучкими порогами. З самого початку було помічено, що мережа таких пристроїв може реалізувати будь-яку булеву

функцію, що легко побачити з того факту, що функції І та АБО можуть бути реалізовані та використані в диз'юнктивній або кон'юнктивальній парадигмі. Дослідники також швидко зрозуміли, що динамічні системи з пам'яттю можна ідентифікувати за допомогою повторюваних мереж зворотного зв'язку нейронів, але більшість досліджень були (і досі) зосереджені на суворо прямих комунікаційних мережах через їх низьку складність.

Однією з важливих та інноваційних штучних нейронних мереж, які використовують лінійні порогові функції, є перцептрон, розроблений Френком Розенблатом. Ця модель розглянула більш гнучкі значення ваги в нейронах і використовувалася в машинах з адаптивними можливостями. Бернард Відроу ввів поріг у 1960 році як термін для відхилення — див. Айдалін.

Наприкінці 1980-х років, коли дослідження нейронних мереж відновили сили, вони почали розглядати нейрони з більш безперервними формами. Можливість диференціювати функцію активації дозволяє безпосередньо використовувати градієнтний спуск та інші алгоритми оптимізації для налаштування ваг. Нейронні мережі також починають використовуватися як загальні моделі для апроксимації функцій. Найвідоміший алгоритм навчання, який називається зворотним поширенням, був перевідкритий кілька разів, але його першу розробку можна простежити до роботи Пола Вербоса [12] [13].

Існує дві парадигми навчання нейронних мереж - з учителем і без вчителя. У першому випадку, на вхідний вектор є готова відповідь, у другому випадку нейронна мережа самонавчається. У кожного виду навчання є своя ніша завдань і за великим рахунком вони не перетинаються. На даний момент придумано і запатентовано велика кількість архітектур нейронних мереж і методів їх навчання. Але основними (вихідними) є - для навчання з учителем це «алгоритм зворотного поширення помилки», а для навчання без учителя це алгоритми Хебба і Кохонена. Ці парадигми сильно перетинаються з біологічної дійсністю, наприклад - дитина навчається з учителем або без?

Також, останнім часом, сформувався нова, третя парадигма - навчання з підкріпленням.

### Навчання з вчителем

Ця категорія навчання має справу в парах  $X$  ' ; і  $Y$  ' ;, і завдання - відобразити їх у функції  $f: X \rightarrow Y$ . Тут  $Y$  дана - це вчитель, планові бажані виходи, і  $X$  дані - це дані, які генерують  $Y$  дані. Це аналогічно вчителю, який навчає всіх виконувати певне завдання.

Одна виняткова особливість цієї парадигми навчання - пряме посилення на помилку, яка просто порівнює між плановим і поточним результатом. Параметри мережі скармливаються в вартісну функцію, яка визначає кількість невідповідності між бажаним і поточним висновком.

Навчання з учителем - дуже придатна для задач, які заздалегідь надає патерн, мета досягнення. Приклади: класифікація картинок, розпізнавання голосу, функція апроксимації, прогнозування. Зверніть увагу, що нейросеть слід забезпечити попередніми знаннями парою вводили-незалежних значень ( $X$ ) і висновком Класифікаци-залежних значень ( $Y$ ). Присутність залежних висновком - обов'язкова умова для навчання з учителем.

### Навчання без учителя

В навчанні без учителя ми маємо справу лише з даними без міток або класифікації; замість цього наша нейронна структура намагається намалювати логічні висновки і витягти знання з цього використовуючи тільки вхідні дані  $X$ .

Це аналогічно самонавчання коли хтось навчає його / її завдяки його / її досвіду і встановлюючи побічні критерії. в навчанні без учителя ми не визначаємо бажаний патерн для кожного спостереження але нейронна структура може навчатися без будь-якого вчителя.

Тут, функція вартості (cost function) грає важливу роль. Вона сильно впливає на значення нейрона так само добре як і зв'язок між вхідними значеннями.

Приклади завдань, де використовується навчання без учителя: кластеризація, компресія даних, статистичні моделі і мовні моделі.

Модель нейронної мережі описує її архітектуру і конфігурацію, а також використовувани алгоритми навчання.

Архітектура нейронної мережі визначає загальні принципи її побудови (плоскошарова, повнозв'язна, слабозв'язана, прямого поширення, рекурентна і т.д.).

Конфігурація конкретизує структуру мережі в рамках заданої архітектури: число нейронів, число входів і виходів мережі, які використовуються активаційні функції.

Розрізняють такі базові архітектури:

1. мережі прямого поширення - всі зв'язки направлені строго від вхідних нейронів до вихідних. До таких мереж відносяться, наприклад персептрон Розенблатта і багатошаровий персептрон;
2. рекурентні нейронні мережі - сигнал з вихідних нейронів або нейронів прихованого шару частково передається назад на нейрони вхідного шару мережі;
3. мережі радіально-базисних функцій - мережі, що містять єдиний прихований шар, нейрони якого використовують радіально-симетричну активаційну функцію, застосовуються для вирішення задач класифікації та прогнозування;
4. мережі Кохонена - клас мереж, що використовують навчання без учителя і призначених для вирішення завдань кластеризації. Вони містять всього два прошарки: вхідний (розподільний) і вихідний (кластеризуються);
5. карти Кохонена або карти ознак, що самоорганізуються - різновид мереж Кохонена, в яких число вихідних нейронів вибирається багато більше числа формованих кластерів. Використовуються для візуалізації результатів кластеризації багатовимірних даних;
6. повнозв'язні мережі - нейронні мережі, в яких кожен нейрон пов'язаний з усіма іншими нейронами. Такі мережі мають найвищу щільність зв'язків;
7. слабозв'язаних мережі - в них нейрони з'єднані тільки зі своїми найближчими сусідами;
8. нейронні мережі з плоскими шарами - в них нейрони утворюють каскади, звані шарами, при цьому нейрони кожного шару пов'язані з усіма

нейронами наступного і попереднього шарів, а всередині шару зв'язків немає. Мережі з плоскими шарами можуть бути одношарові (містити один прихований шар) і багатшаровими (містити кілька прихованих шарів).

Кожна архітектура мережі призначення для вирішення певного класу задач аналізу даних (регресії, класифікації, кластеризації, прогнозування) і використовує спеціальні алгоритми навчання.

### Навчання

Звичайна критика нейронних мереж, особливо в робототехніці, полягає в тому, що вони вимагають занадто багато навчання для роботи в реальному світі. Потенційні рішення включають випадкове перемішування навчальних прикладів за допомогою алгоритму чисельної оптимізації, який не робить занадто великих кроків, коли під час роботи не виконується занадто великий крок. зміна мережевих підключень за прикладом, групування прикладів у так звані міні-пакети та/або впровадження рекурсивного алгоритму найменших квадратів для СМАС.

Основне заперечення полягає в тому, що ШНМ недостатньо відображають функцію нейронів. Зворотне поширення є критичним кроком, хоча в біологічних нейронних мережах такого механізму не існує. Як інформація кодується реальними нейронами, невідомо. Сенсорні нейрони частіше запускають потенціали дії при активації сенсора, а м'язові клітини тягнуться сильніше, коли пов'язані з ними моторні нейрони частіше отримують потенціали дії. Крім випадку передачі інформації від сенсорного нейрона до моторного нейрона, майже нічого з принципів того, як інформація обробляється біологічними нейронними мережами, не відомо.

Основна вимога ШНМ полягає в тому, що вони втілюють нові потужні загальні принципи обробки інформації. Ці принципи погано визначені. Часто стверджують, що вони походять із самої мережі. Це дозволяє описувати просту статистичну асоціацію (основну функцію штучних нейронних мереж) як навчання або розпізнавання. У 1997 році Олександр Дьюдні прокоментував, що в результаті штучні нейронні мережі мають якість «щось за ніщо», таку, яка надає

особливу ауру лінощів і чіткої відсутності цікавості щодо того, наскільки хороші ці обчислювальні системи. Жодна людина втручається рука (чи розум), рішення знаходять ніби за допомогою чарівної палички, а ніхто, здається, нічого не навчився». Однією з реакцій Дьюдні є те, що нейронні мережі справляються з багатьма складними та різноманітними завданнями, починаючи від автономного польоту літака і закінчуючи виявленням шахрайства з кредитними картками до оволодіння грою Go.

До теперішнього часу найбільшого поширення набули такі види нейромереж: - згорткові нейронні мережі (CNN) імітують роботу зорової кори головного мозку і частково виконують функцію абстрактного мислення. Вони прекрасно справляються із завданням розпізнавання зображень, а їх обчислення легко распаралелить на графічних процесорах, що дозволяє створювати відносно дешеві апаратні платформи з елементами ШІ.

CNN застосовуються в системах машинного зору безпілотних автомобілів, комерційних дронів, роботів, а також в охоронному відеоспостереженні. Ви теж кожен день використовуєте CNN, якщо включили в налаштуваннях смартфона розблокування за допомогою розпізнавання особи.

- рекурентні нейронні мережі (RNN) володіють короткочасною пам'яттю, за рахунок чого легко аналізують послідовності довільної довжини. RNN розбивають потік даних на елементарні частини і оцінюють взаємозв'язки між ними. Ці алгоритми знайшли основне застосування в розпізнаванні рукописного тексту й мови. Коли ви шукаєте мелодію на слух в Shazam, розмовляєте з Siri, Google Now або Алісою від «Яндекса», залишаєте замітки від руки для Cortana - на хмарних платформах беруться за справу рекурентні нейромережі.

- мережі з довготривалою і короткочасною пам'яттю (LSTM) стали подальшим розвитком RNN. Вони гарні для прогнозування змін будь-якої величини (наприклад, біржових курсів або купівельного попиту) шляхом екстраполяції. Також їх застосовують для глибокого аналізу природної мови. Наприклад, Google використовує LSTM в персональному помічника і системі машинного перекладу Google Translate. Без LSTM якість перекладів так і

залишалося б на рівні програм з дев'яностих, з якими часом було простіше перевести текст самому, ніж виправляти численні помилки.

- керовані рекурентні блоки (GRU) - порівняно недавня модифікація RNN, що з'явилася тільки в 2014 році. Їх використовують для синтезу мови, яка володіє емоційним забарвленням і звучить як справжня. Наприклад, в тестах сервісів Google Duplex і Microsoft Xiaoice люди не змогли відрізнити говорять ботів від живих співрозмовників. Примітно, що Xiaoice дозволила Microsoft зміцнитися на азіатському ринку, де розвиток компанії завжди стримувалося мовним бар'єром.

- глибокі нейронні мережі (DNN) - будь-яка мережа більш ніж з трьома шарами. Вони лежать в основі механізмів глибокого машинного навчання, знаходячи неявні взаємозв'язку між різнорідними даними. Яскравий приклад - пошук кореляцій між розвитком захворювань і різними потенційними факторами у величезних масивах наукових статей за допомогою IBM Watson.

- генеративно-змагальні мережі (GAN). Це комбінація нейромереж, одна з яких генерує варіанти, а інша відсіває їх (виступає в якості арбітра). Таке поєднання дозволяє реалізувати машинне навчання без учителя, що підвищує автономність П. Наприклад, PixelDTGAN генерує окремі зображення одягу, взуття та аксесуарів для каталогів онлайн-магазинів. В якості вхідних даних використовуються фотографії, на яких ці предмети гардероба демонструють фотомоделі. Поки зйомка для каталогів одягу вважається досить витратною частиною електронної комерції, але цілком можливо, що в найближчі роки нейромережі дозволять швидко проілюструвати каталог, навіть не залучаючи фотографа. Обробка фотографій теж забирає багато часу. Ви можете спробувати зробити це набагато швидше, додаючи і прибираючи графічні об'єкти за допомогою іншої нейромережі - IBM GANPaint. Подібна їй нейросеть DRAGAN вже застосовується для автоматичної відтворення персонажів аніме і мультфільмів. Вона дозволяє прискорити вихід нових серій і утримати аудиторію розважальних каналів, не перевантажуючи аніматорів колосальним об'ємом роботи. Так що там мультфільми! GAN дозволяють анімувати

тривимірну модель людини, переносячи на неї руху іншого в реальному часі. Перші результати виглядають не дуже переконливо, однак особливість будь нейромережі - в тому, що вона покращується з кожною новою порцією даних.

#### **1.4 Постановка задачі**

Провівши аналіз предметної області можна зробити висновок, що існуючи на даний час системи автоматизації забезпечення екологічної рівноваги у сфері переробки відходів не використовують оптимізаційної моделі забезпечення підтримки екологічної рівноваги.

Для розв'язання визначеної проблеми необхідно розробити оптимізаційну модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

**Мета і задачі роботи** полягає у розробці оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Для досягнення поставленої мети визначені наступні задачі дослідження:

- побудова моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів;
- для дослідження практичної ефективності розробити інформаційну модель системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

#### **Висновки до розділу 1**

В першому розділі проведено аналіз існуючих характеристик предметної області з напрямку кваліфікаційної роботи. Спираючись на отримані результати аналізу проведена систематизація і отриманий аналіз потреб до моделі інформаційної системи, що створюється.

Визначено мету і задачі роботи яка полягає у розробці оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів

## **Розділ 2**

### **Розробка інформаційної технології для забезпечення підтримки екологічної рівноваги**

#### **2.1 Створення моделі оптимізації переробки відходів**

Аналізуючи існуючі сервіси прийняття рішень в різноманітних сферах виявлено, що сфера переробки відходів не містить в собі автоматизаційних рішень, окрім тих, що допомагають автоматизувати безпосередньо сам процес переробки відходів (наприклад, різного роду автоматичні конвеєри, тощо).

Виходячи з відсутності рішень, які виконували б функціонал прийняття рішень стосовно вибору методу переробки відходів необхідно розробити дану систему.

Система повинна складатися з таких кроків (рисунок 2.1):

- вхідна інформація: відповіді користувача на питання стосовно характеристик відходів;
- обробка отриманої інформації і передача до моделі нейронної мережі;
- обробка отриманих даних нейронною мережею і передача отриманого результату;
- вихідні дані: метод переробки відходів.

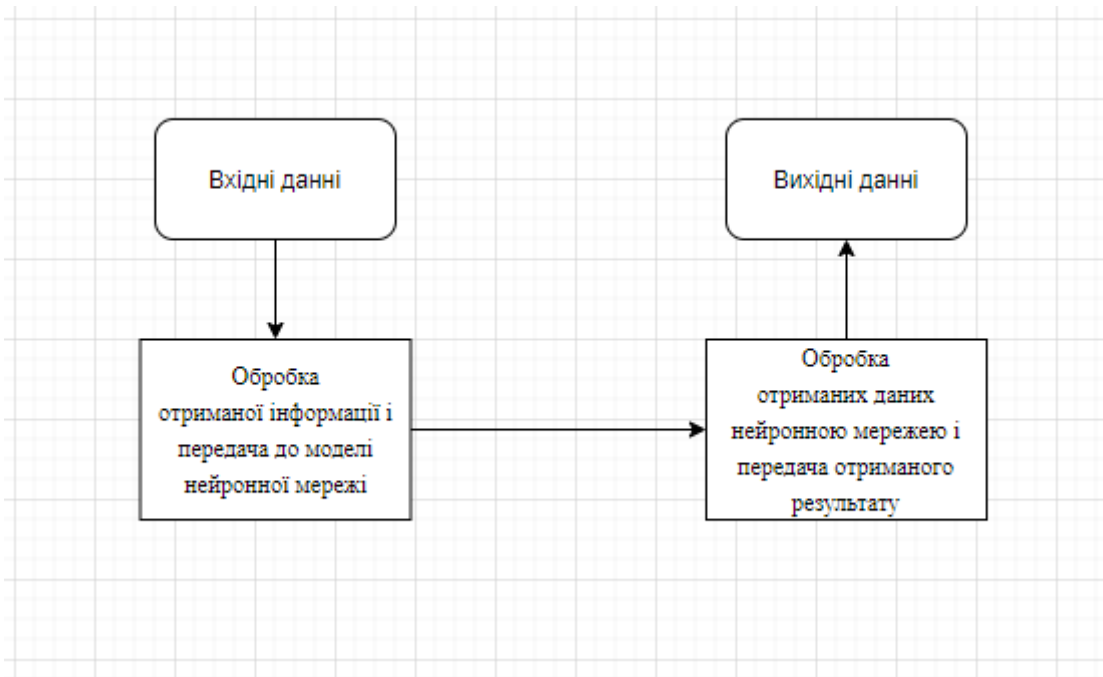


Рисунок 2.1 – Схема роботи моделі оптимізації переробки відходів

Оскільки модель базується на прийнятті рішень стосовно технологій переробки відходів слід розглянути існуючі технології, які в подальшому будуть включені в модель.

## 2.2 Побудова моделі підтримки екологічної рівноваги

Математична модель — це опис системи за допомогою математичних понять і мови. Процес встановлення математичної моделі називається математичним моделюванням. Математичні моделі використовуються в природничих науках (наприклад, фізика, біологія, науки про Землю, хімія) та інженерних дисциплінах (наприклад, інформатика, електротехніка) і нефізичних системах, таких як соціальні науки (наприклад, економіка), психологія, Соціологія, політика). Використання математичних моделей для вирішення завдань комерційних або військових операцій є важливою частиною галузі дослідження операцій. Математичні моделі також використовуються в музиці, [1] лінгвістиці [2] та філософії (наприклад, зосереджені в аналітичній філософії).

Модель може допомогти пояснити систему та вивчити вплив різних компонентів, а також передбачити поведінку.

Математичні моделі можуть приймати різні форми, включаючи динамічні системи, статистичні моделі, диференціальні рівняння або теоретичні ігрові моделі. Ці та інші типи моделей можуть перекриватися, і модель включає різні абстрактні структури. Як правило, математична модель може включати логічну модель. У багатьох випадках якість наукової галузі залежить від узгодженості математичної моделі теоретичної розробки та результатів повторних експериментів. У міру розвитку кращих теорій відсутність узгодженості між теоретичними математичними моделями та експериментальними вимірюваннями часто призводить до важливих успіхів.

У фізичних науках традиційні математичні моделі включають більшість наступних елементів:

- керівні рівняння;
- додаткові підмоделі;
- визначення рівнянь;
- установчі рівняння;
- припущення та обмеження;
- початкові та граничні умови;
- класичні обмеження та кінематичні рівняння.

Математичні моделі зазвичай складаються із зв'язків і змінних. Відношення можуть бути описані такими операторами, як алгебраїчні оператори, функції та диференціальні оператори. Змінні — це абстракції цікавих системних параметрів, які можна кількісно оцінити. Для математичних моделей, розділених за структурою, можна використовувати різноманітні критерії класифікації:

Лінійна або нелінійна: якщо всі оператори в математичній моделі демонструють лінійність, отримана математична модель визначається як лінійна. В іншому випадку модель вважається нелінійною. Визначення лінійного та нелінійного залежить від контексту, а лінійні моделі можуть мати нелінійні

вирази. Наприклад, статистична лінійна модель передбачає, що зв'язок є лінійною за параметрами, але може бути нелінійною з точки зору предикторів. Аналогічно, якщо диференціальне рівняння можна записати за допомогою лінійних диференціальних операторів, воно називається лінійним, але воно все одно може мати нелінійні вирази. У моделі математичного програмування, якщо цільова функція та обмеження повністю представлені системою лінійних рівнянь, модель вважається лінійною моделлю. Якщо одна або кілька цільових функцій або обмежень представлені нелінійним рівнянням, модель називається нелінійною моделлю.

Проблему припущення лінійної структури можна розкласти на простіші частини, які можна розглядати незалежно та/або аналізувати в різних масштабах. Результати залишаються справедливими для початкової задачі рекомбінації та масштабування.

Нелінійність, навіть у досить простих системах, часто пов'язана з такими явищами, як хаос і незворотність. Хоча є винятки, нелінійні системи та моделі, як правило, складніше вивчати, ніж лінійні системи та моделі. Поширеним методом для нелінійних задач є лінеаризація, але при спробі вивчити незворотність, тісно пов'язану з нелінійністю, це може викликати проблеми.

Статична і динамічна: динамічна модель розглядає зміни стану системи з часом, тоді як статична (або статична) модель розраховує, що система перебуває у збалансованому стані, тому вона є інваріантною в часі. Динамічні моделі зазвичай виражаються диференціальними рівняннями або різницевиими рівняннями.

Явні та неявні: якщо всі вхідні параметри загальної моделі відомі, і для обчислення вихідних параметрів можна використовувати кінцеву серію обчислень, модель називається явною. Але іноді початкові параметри відомі, і відповідні вхідні дані необхідно розв'язувати за допомогою ітераційного процесу, такого як метод Ньютона або метод Бroyдена. У цьому випадку модель називається неявною. Наприклад, на основі розрахованого теплового циклу (потік повітря та палива, тиск і температура) за конкретних умов польоту та

налаштувань потужності можна точно розрахувати фізичні характеристики реактивних двигунів, такі як площа турбіни та сопла, Однак робочий цикл двигуна і параметри потужності в інших умовах польоту неможливо точно розрахувати на основі постійних фізичних характеристик.

Дискретні та безперервні: дискретні моделі розглядають об'єкти як дискретні, наприклад, частинки в молекулярних моделях або стани в статистичних моделях; у той час як безперервні моделі представляють об'єкти безперервно, такі як поле швидкості рідини в потоці труби, температура та напруження в твердих тілах Поле, а також електричне поле, безперервно діє на всю модель через точкові заряди.

Детермінізм і ймовірність (випадкові): детермінована модель - це модель, в якій кожен набір станів змінних однозначно визначається параметрами моделі та набором попередніх станів цих змінних; отже, для даного набору початкових умов, визначина модель завжди однакова. Навпаки, у випадковій моделі, яку зазвичай називають «статистичною моделлю», є випадковість, і змінні описуються неунікальними значеннями, а скоріше розподілом ймовірностей.

Дедукція, індукція або float: модель дедукції — це логічна структура, заснована на теорії. Індуктивні моделі виводяться на основі емпіричних висновків і узагальнень на їх основі. Плаваюча модель базується не на теорії чи спостереженнях, а лише на виклику очікуваній структурі. Використання математики в соціальних науках, крім економіки, критикувалося як необґрунтована модель [3]. Застосування теорії катастроф у науці описується як плаваюча модель [4].

Різниця між стратегічними і нестратегічними моделями, що використовуються в теорії ігор, полягає в тому, що вони моделюють агентів з несумісними стимулами, наприклад, конкуруючих видів або аукціоністів. Стратегічна модель передбачає, що учасники є автономними особами, які приймають рішення, і вони раціонально вибирають дії, які максимізують їх цільові функції. Ключовим питанням використання стратегічних моделей є визначення та розрахунок концепцій прийняття рішень, таких як рівновага

Неша.Цікавою особливістю стратегічних моделей є те, що вони відокремлюють розгляд правил гри від розгляду поведінки гравця [5].

У бізнесі та технологіях математичні моделі можна використовувати для максимізації конкретних результатів. Розглянута система зажадає певних вхідних даних. Система, яка пов'язує вхідні дані з вихідними, також залежить від інших змінних: змінних рішення, змінних стану, екзогенних змінних і випадкових величин.

Розв'язки змінних іноді називають незалежними змінними. Екзогенні змінні іноді називають параметрами або константами. Змінні не є незалежними одна від одної, оскільки змінні стану залежать від рішення, вхідних, випадкових та екзогенних змінних. Крім того, вихідна змінна залежить від стану системи (представлена змінною стану).

Цілі та обмеження системи та її користувачів можуть бути виражені як функція вихідних змінних або змінних стану. Цільова функція буде залежати від погляду користувача на модель.Залежно від контексту цільову функцію також називають показником ефективності, оскільки вона є конкретним показником, який цікавить користувачів. Хоча кількість цільових функцій і обмежень, які може мати модель, не обмежена, у міру збільшення кількості використання або оптимізація моделі ускладнюється (обчислювальною).

Наприклад, економісти часто використовують лінійну алгебру при використанні моделей «вхід-випуск». Вектори можна використовувати для поєднання складних математичних моделей з багатьма змінними, де один символ представляє кілька змінних.

Завдання математичного моделювання зазвичай діляться на моделі чорного ящика або моделі білого ящика, залежно від того, наскільки доступна попередня інформація про систему. Модель чорного ящика – це система без попередньої інформації. Модель білого ящика (також називається скляною коробкою або прозорою коробкою) — це система, в якій доступна вся необхідна інформація.Майже всі системи знаходяться між моделлю чорного ящика і

моделлю білого ящика, тому ця концепція використовується лише як інтуїтивне керівництво для методу вибору.

Зазвичай краще використовувати якомога більше попередньої інформації, щоб зробити модель точнішою. Тому модель білого ящика, як правило, вважається легкою, тому що якщо ви правильно використовуєте інформацію, модель буде функціонувати правильно. Зазвичай апріорна інформація надходить у формі знання про типи функцій, пов'язаних з різними змінними. Наприклад, якщо ми змодельємо дію ліків на людську систему, ми будемо знати, що кількість ліків у крові зазвичай є функцією, яка зменшується в геометричній прогресії. Але у нас є ще невідомі параметри: як швидко розпадається препарат? Яка початкова кількість ліків у крові? Тому цей приклад не є повністю білою моделлю. Перед використанням моделі ці параметри необхідно якимось чином оцінити.

Модель чорного ящика намагається оцінити функціональну форму зв'язку між змінними та числовими параметрами в цих функціях. Наприклад, використовуючи апріорну інформацію, ми можемо отримати набір функцій, які можуть адекватно описати систему. Якщо попередньої інформації немає, ми спробуємо охопити всі різні моделі максимально загальними функціями. Поширеним методом моделі чорного ящика є нейронна мережа, яка зазвичай не робить припущень щодо вхідних даних. Крім того, розроблений в рамках ідентифікації нелінійної системи алгоритм NARMAX (модель нелінійної авторегресійної ковзної середньої з екзогенним входом [6]) можна використовувати для вибору елементів моделі, визначення структури моделі та оцінки наявності кореляції та нелінійності. Шум невідомого параметра. Перевага моделі NARMAX перед нейронною мережею полягає в тому, що модель, створена NARMAX, може бути записана та пов'язана з основним процесом, тоді як нейронні мережі створюють наближення, яке є непрозорим.

#### Суб'єктивна інформація

Іноді буває корисно включати суб'єктивну інформацію в математичні моделі. Це можна зробити на основі інтуїції, досвіду чи думки експертів, або на

основі зручності математичних форм. Байєсівська статистика забезпечує теоретичну основу для включення цієї суб'єктивності в суворий аналіз: ми визначаємо попередній розподіл ймовірностей (який може бути суб'єктивним), а потім оновлюємо розподіл на основі емпіричних даних.

Прикладом необхідності цього методу є коли експериментатор злегка згинає монету і кидає її один раз, щоб визначити, чи вона впала з головки, а потім отримує завдання передбачити ймовірність того, що монета впаде з головки наступного разу. монету підкидають. Після згинання монета монета впаде з головки. Справжня ймовірність летіти в лоб невідома, тому експериментатор повинен вирішити (можливо, спостерігаючи за формою монети), який попередній розподіл до використання. Включення такої суб'єктивної інформації може бути важливим для отримання точних оцінок ймовірності.

Будь-яка модель не білого ящика містить деякі параметри, які можна використовувати для адаптації моделі до системи, яку вона описує. Якщо для моделювання використовуються штучні нейронні мережі або інше машинне навчання, оптимізація параметрів називається навчанням, а оптимізація гіперпараметрів моделі називається налаштуванням, і зазвичай використовується перехресна перевірка [8]. У більш традиційному моделюванні з використанням чітко визначених математичних функцій параметри зазвичай визначаються підгоночними кривими.

Зазвичай найпростіша частина оцінки моделі - це перевірити, чи відповідає модель експериментальним вимірюванням або іншим емпіричним даним. У моделі з параметрами загальний спосіб перевірки відповідності полягає в поділі даних на дві непересічні підмножини: навчальні дані та дані перевірки. Навчальні дані використовуються для оцінки параметрів моделі. Точна модель точно відповідатиме тестовим даним, навіть якщо дані не використовуються для встановлення параметрів моделі. Така практика в статистиці називається перехресною перевіркою.

Визначення показників, які вимірюють відстань між спостережуваними даними та прогнозованими даними, є корисним інструментом для оцінки відповідності моделі. У статистиці, теорії рішень і деяких економічних моделях функція втрат відіграє подібну роль.

Хоча легко перевірити відповідність параметрів, може бути складніше перевірити загальну математичну форму моделі. Взагалі кажучи, порівняно з моделями, що включають диференціальні рівняння, було розроблено більше математичних інструментів для перевірки узгодженості статистичних моделей. Іноді можна використовувати непараметричні статистичні інструменти, щоб оцінити, наскільки добре дані відповідають відомому розподілу, або створити загальну модель, яка робить лише мінімальні припущення щодо математичної форми моделі.

Оцінити область застосування моделі, тобто визначити, де модель застосовна, може бути не так просто. Якщо модель заснована на наборі даних, необхідно визначити, які системи або умови мають відомі дані, які є «типовими» наборами даних.

Проблема того, чи добре модель описує атрибути системи між точками даних, називається інтерполяцією, а така ж проблема для подій або точок даних за межами даних спостереження називається екстраполяцією.

Як приклад типових обмежень моделі при оцінці класичної механіки Ньютона можна відзначити, що Ньютон проводив свої вимірювання без сучасного обладнання, тому він не міг виміряти характеристики частинок, що рухаються зі швидкістю, близькою до швидкості світла. Так само він не вимірював рух молекул та інших дрібних частинок, а вимірював лише великі частинки. Тому, хоча його моделі достатньо для звичайної фізики в житті, не дивно, що його модель погано виводить ці області. Багато типів моделювання неявно передбачають твердження про причинно-наслідковий зв'язок. Зазвичай (але не завжди) це правильно для моделей, що включають диференціальні рівняння. Оскільки метою моделювання є розширити наше розуміння світу, ефективність моделі залежить не лише від її відповідності емпіричним

спостереженням, а й від її здатності робити висновки про ситуації чи дані за межами того, що модель спочатку описувала. Це можна розглядати як різницю між якісним і кількісним прогнозуванням. Можна також стверджувати, що модель марна, якщо вона не дає розуміння відомого знання за межі безпосередньо досліджуваного явища.

Прикладом такої критики є те, що математична модель теорії оптимального виробництва харчових продуктів не забезпечує розуміння розумних висновків за межі основних принципів еволюції та інших екологічних принципів [9].

Основна задача проектованої моделі полягає в прийнятті рішень стосовно вибору методу переробки відходів. Для створення моделі слід виділити основні критерії, які будуть фігурувати в ній.

Для побудови моделі було обрано наступні критерії:

- можливість участі у рециклінгу;
- коефіцієнт горючості;
- наявність в складі металів;
- відношення до органічних відходів.

Оскільки математична модель здатна оброблювати виключно числові данні — було прийнято рішення використовувати числові еквіваленти булевих значень «так» і «ні», тобто нуль і одиницю.

Кожний із параметрів приймає одне з оговорених значень, тобто «0» або «1» в залежності від характеристик відходів. Наприклад, якщо відходи горять у вогні — параметр горючості прийме значення «1».

Дані параметри складають фінальний результат, який приймає значення від 1 до 3, по кількості обраних для моделі методів переробки відходів.

В первинній формі модель прийняття рішення виглядає наступним чином:

$$N_{meth} = k_{rec} + k_{flam} + k_{met} + k_{org}, \quad (2.1)$$

де  $N_{meth}$  — порядковий номер методу переробки відходів (1 — рекуперація, 2 — рециклінг, 3 — остаточна утилізація),

$k_{rec}$ — числовий еквівалент можливості участі у рециклінгу,

$k_{flam}$ — числовий еквівалент горючості,

$k_{met}$ — числовий еквівалент наявності металів у складі,

$k_{org}$ — числовий еквівалент відношення до органічних відходів.

Проте, дана модель не має необхідної гнучкості, що може призвести до отримання невірних результатів процесу прийняття рішень. Для виключення даного фактору слід включити у модель ситуативні коефіцієнти, які будуть відповідати за важливість кожного фактору в кожному конкретному випадку використання моделі і будуть підбиратися в ході використання.

Після додавання ситуативних коефіцієнтів важливості фінальний вигляд математичної моделі виглядає наступним чином:

$$N_{meth} \approx w_{rec} * k_{rec} + w_{flam} * k_{flam} + w_{met} * k_{met} + w_{org} * k_{org}, \quad (2.2)$$

де  $N_{meth}$  — порядковий номер методу переробки відходів (1 — рекуперація, 2 — рециклінг, 3 — остаточна утилізація),

$k_{rec}$ — числовий еквівалент можливості участі у рециклінгу,

$k_{flam}$ — числовий еквівалент горючості,

$k_{met}$ — числовий еквівалент наявності металів у складі,

$k_{org}$ — числовий еквівалент відношення до органічних відходів,

$w_{rec}$ — ситуативний коефіцієнт важливості параметру можливості участі у рециклінгу,

$w_{flam}$ — ситуативний коефіцієнт важливості параметру горючості,

$w_{met}$ — ситуативний коефіцієнт важливості параметру наявності металів у складі,

$w_{org}$ — ситуативний коефіцієнт важливості параметру відношення до органічних відходів.

Отримана модель має можливість тонкого налаштування за рахунок зміни значень коефіцієнтів важливості, що надає можливість отримувати більш точні результати. Окрім додавання ситуативних коефіцієнтів важливості слід звернути увагу на те, що у (2.2) «тверда рівність» методу і суми параметрів змінилась на наближену рівність, тобто округлення, що дозволяє отримати результат при

нецілому результати розрахунків, наприклад результат 2.66 буде однозначно округлений до 3, що означає третій метод переробки відходів, а саме — остаточну утилізацію.

### **2.3 Аналіз вимог до інформаційної моделі системи з підтримки екологічної рівноваги у сфері переробки відходів**

Оскільки побудова моделі автоматизованого процесу без практичного впровадження не несе користі, було прийнято рішення на основі моделі створити нейронну мережу, яка б, в свою чергу, реалізувала б основні принципи і функціональні можливості моделі у вигляді системи прийняття рішень.

В свою чергу проектування будь-якої системи вимагає створення переліку вимог до системи. Нижче приведено вимоги до розроблюваної системи, яка базується на моделі прийняття рішень стосовно вибору методу переробки відходів.

IEEE (Institute of Electrical and Electronics Engineers – Інститут інженерів електротехники и електроніки) Standard Glossary of Software Engineering Terminology визначає вимоги як:

- умови або можливості, необхідні користувачеві для вирішення проблем або досягнення цілей;
- умови або можливості, якими повинна володіти система або системні компоненти, щоб виконати контракт або задовольняти стандартам, специфікаціям або іншим формальним документам;

Бізнес – правило (business rule) – політика, припис, стандарт, правило або обчислювальна формула, яка визначає чи обмежує деякі сторони бізнес – процесів.

Бізнес – вимоги (business requirement) – обсяг інформації, який в сукупності описує потреба, яка ініціює один або більше проектів, покликаних надати рішення і отримати необхідний кінцевий бізнес – результат. Бізнес –

вимоги включають бізнес – можливості, бізнес – цілі, метрики успіху, концепція і кордони і обмеження.

Обмеження (constraint) – накладається на доступні розробнику можливості дизайну або конструювання продукту. Інші типи обмежень можуть обмежити можливості, доступні для менеджерів проектів. Бізнес – правила часто накладають обмеження на бізнес – операції, а значить, на програмні системи.

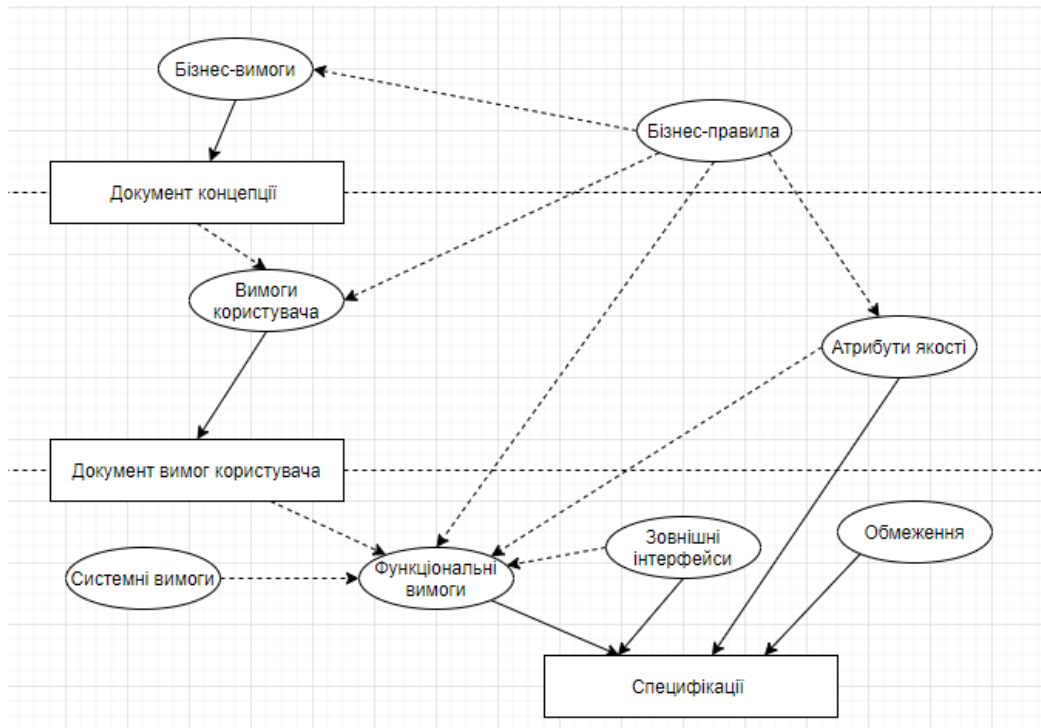


Рисунок 2.2 Взаємозв'язки декількох типів інформації для вимог

Функціональна вимога (functional requirement) – опис поведінки системи в певних умовах.

Атрибут якості (quality attribute) – вид нефункціонального вимоги, що описує характеристику сервісу або продуктивності продукту. Приклади атрибутів якості: зручність і простота використання, легкість переміщення, легкість в експлуатації, цілісність, надійність, ефективність і стійкість до збоїв. У вимогах описані рамки атрибутів якості, до яких продукт демонструє бажані характеристики.

Користувацька вимога (user requirement) – мета і завдання, яку користувачі повинні мати можливість виконувати з системою, або положення

про очікування користувачів про якість системи. Призначені для користувача вимоги зазвичай подаються у вигляді варіантів використання, призначених для користувача історій і сценаріїв.

Системне вимога (system requirement) – вимога верхнього рівня до продукту, що складається з багатьох підсистем, які можуть являти собою ПЗ або сукупність ПЗ і устаткування[14].

Вимоги для інтерфейсу зовнішнього пристрою (external interface requirement) – опис інтерфейсу між системою ПЗ і користувачем, іншою системою ПЗ або обладнанням.

Нефункціональна вимога (nonfunctional requirement) – опис властивих властивостей або характеристик, які система ПЗ повинна демонструвати, або обмеження, які вона повинна дотримуватися[14].

Вимоги до системи інтернет-магазину за різними типами інформації наведено у таблиці 2.1.

Таблиця 2.1 – Вимоги до системи

Тип вимоги	Вимоги до системи
Бізнес – правило	Неможливо запустити процес прийняття рішень без відповідей на питання системи. Неможливо запустити процес прийняття рішень, давши відповіді у невірному форматі Неможливо запустити процес прийняття рішень без відповіді на всі питання, поставлені системою
Функціональні вимоги	Контроль формату відповідей на питання Контроль наявності відповідей на питання
Системні вимоги	Система повинна підтримувати інструментальні засоби, використані при розробці

Вимоги користувачів	Доступ до системи; Ініціювання процесу прийняття рішень; Отримання відповіді у зрозумілому форматі.
Нефункціональні вимоги	Повинна підтримуватись операційними системами сімейства Windows та UNIX.
Обмеження	Інтерфейс повинен бути інтуїтивно зрозумілим Питання повинні бути зрозумілими будь-якому користувачеві.
Атрибут якості	Система повинна контролювати наявність і коректність відповідей користувача.

## Висновки до розділу 2

В ході написання другого розділу роботи було проаналізовано технології і методи переробки сміття, проведено загальний аналіз і опис інформаційної технології, виявлено і задокументовано основні вимоги до розроблюваної інформаційної системи, побудовано математичну модель підтримки екологічної рівноваги.

Вся отримана в ході написання розділу інформація ляже в основу розробки і дасть загальне розуміння технологічного забезпечення процесу розробки інформаційної системи з підтримки екологічної рівноваги у сфері переробки відходів.

## Розділ 3

### Проектування реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж

#### 3.1 Основні вимоги до проектування інформаційної системи

Первинним етапом проектування будь-якої системи є аналіз мети проекту і формування вимог до функціоналу розроблюваної системи.

Основною метою створюваної системи є автоматизація процесу вибору методу переробки відходів, що допоможе спростити даний процес.

Основні функціональні вимоги до системи можна представити у вигляді наступного переліку:

- отримання від користувача числових еквівалентів параметрів математичної моделі в зрозумілому для користувача форматі;
- обробка отриманих даних і приведення їх у вигляд, з яким здатна працювати нейронна мережа;
- прийняття рішення стосовно вибору методу переробки відходів нейронною мережею та передача на модуль взаємодії з користувачем;
- виведення отриманого результату в зрозумілій для користувача формі.

#### 3.2 Проектування загальної схеми інформаційної системи

З метою забезпечення виконання функціональних вимог слід створити функціональну модель, яка містить в собі декілька базових функціональних блоків. Модель представлена на рисунку 3.1.

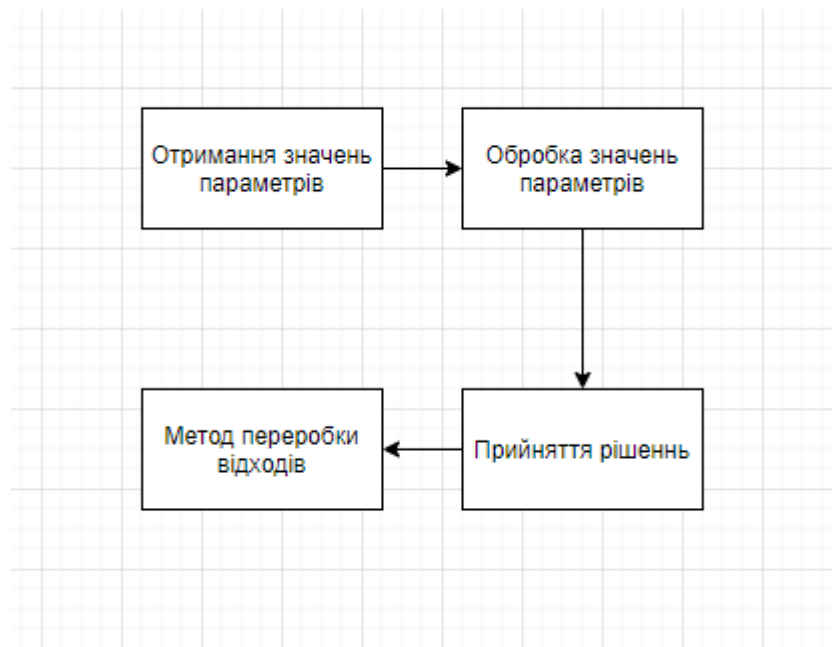


Рисунок 3.1 — Функціональна модель системи

Блок отримання значень параметрів функціонує через інтерфейс користувача і збирає від користувача необхідні дані, а саме числові еквіваленти параметрів математичної моделі через формат взаємодії «питання-відповідь» і передає отримані дані модулю обробки значень.

Модуль обробки значень відповідає за переробку отриманих від клієнта значень параметрів математичної моделі в формат, прийнятний для роботи нейронної мережі, а саме у формат масиву даних.

Блок прийняття рішень після отримання масиву даних розпаковує заздалегідь навчену і запаковану модель нейронної мережі і передає на її вхід масив даних, отримує значення результату і передає його на вихід користувачеві, попередньо замінивши числовий еквівалент методу переробки відходів на його назву.

Створена функціональна модель описує процес обробки інформації, що подається на вхід системи, суть роботи кожного функціонального блоку, що дає

можливість створити власну реалізацію створеної раніше математичної моделі вибору методу переробки відходів.

### **3.3 Проектування інтерфейсної частини системи**

Оскільки проєктована система більше підходить для інтеграції у внутрішню систему підприємства або ресурсу (наприклад, веб-сайту волонтерів, що займаються еко-активізмом) було прийнято рішення про створення демонстраційного інтерфейсу у вигляді командного рядка, що дозволило розподілити навантаження по розробці в бік функціональних вимог, оскільки суттєво скоротився час на проектування інтерфейсу.

Інтерфейс командного рядка (CLI) обробляє команди для комп'ютерної програми у вигляді рядків тексту. Програма, яка обробляє інтерфейс, називається інтерпретатором командного рядка або процесором командного рядка. Операційні системи реалізують інтерфейс командного рядка в оболонці для інтерактивного доступу до функцій або служб операційної системи. Такий доступ в основному надавався користувачам комп'ютерними терміналами, починаючи з середини 1960-х років, і продовжував використовуватися протягом 1970-х і 1980-х років у системах VAX/VMS, Unix та персональних комп'ютерних системах, включаючи DOS, CP/M та Apple DOS.

Сьогодні багато користувачів покладаються на графічні інтерфейси користувача та взаємодію, керовану меню. Однак деякі завдання програмування та обслуговування можуть не мати графічного інтерфейсу користувача і все одно можуть використовувати командний рядок.

Альтернативи інтерфейсу командного рядка включають текстові меню інтерфейсу користувача (наприклад, IBM AIX SMIT), комбінації клавіш і різні метафори на робочому столі з центром вказівника (зазвичай керуються за допомогою миші). Прикладами цього є Microsoft Windows, DOS Shell і PowerPanel системи миші. Інтерфейси командного рядка часто реалізуються в термінальних пристроях, які також можуть використовувати екранно-орієнтовані

текстові інтерфейси користувача, які використовують адресацію курсора для розміщення символів на екрані дисплея.

Програми з інтерфейсом командного рядка, як правило, легше автоматизувати за допомогою сценаріїв.

Багато програмних систем реалізують інтерфейси командного рядка для керування та роботи. Сюди входять середовища програмування та допоміжні програми.

CLI використовується, коли великий словник команд або запитів у поєднанні з широким (або довільним) діапазоном параметрів може бути введений швидше як текст, ніж за допомогою чистого графічного інтерфейсу. Зазвичай це відбувається з командними оболонками операційної системи. CLI також використовуються системами з недостатніми ресурсами для підтримки графічного інтерфейсу користувача. Деякі комп'ютерні мовні системи (такі як Python, Forth, LISP, Rexx і багато діалектів BASIC) забезпечують інтерактивний режим командного рядка для швидкої оцінки коду.

CLI часто використовуються програмістами та системними адміністраторами, в інженерних та наукових середовищах, а також технічно досвідченими користувачами персональних комп'ютерів. CLI також популярні серед людей з вадами зору, оскільки команди та відповіді можуть відображатися за допомогою дисплеїв Брайля, що оновлюються.

Приклад інтерфейсу командного рядка зображено на рисунку 3.2.

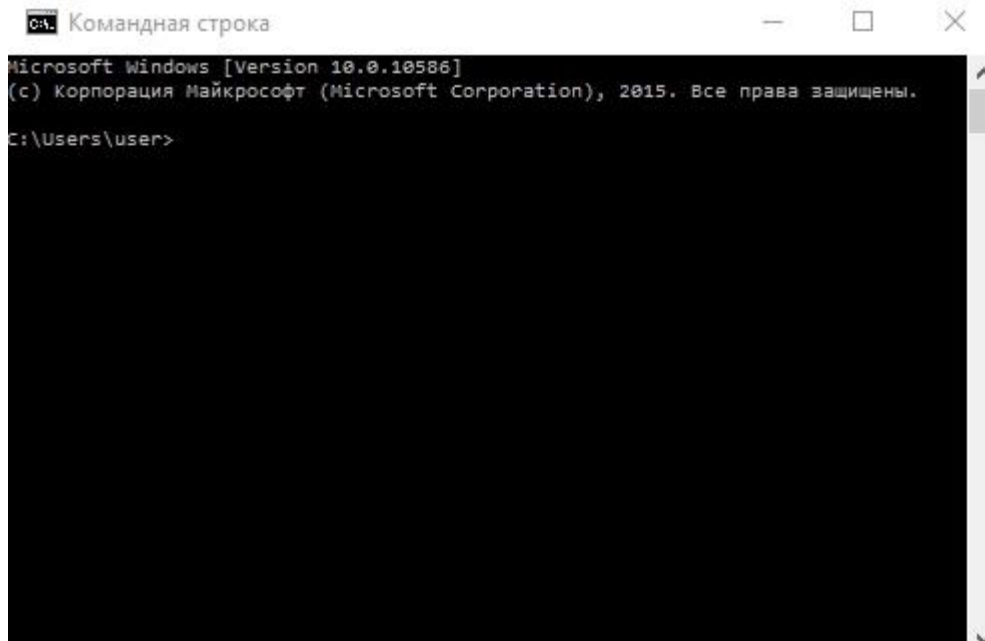


Рисунок 3.2 — Приклад командного рядка

Інтерфейс розроблюваної системи буде базуватися на шаблоні отримання даних, оскільки принцип отримання усіх даних є однаковим. Шаблон інтерфейсу взаємодії користувача з системою показано на рисунку 3.3.

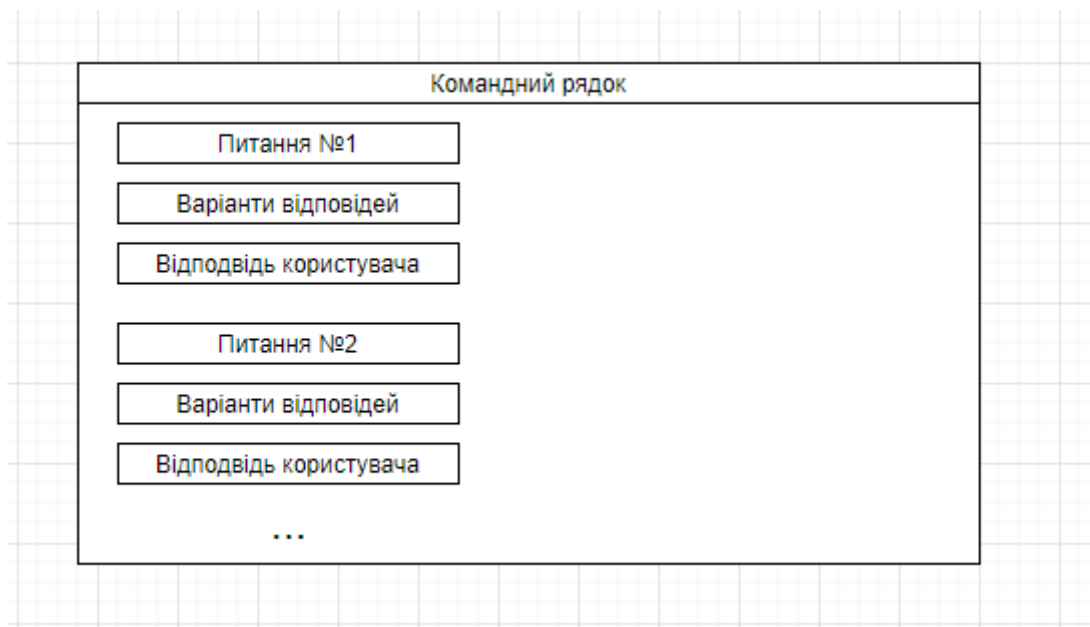


Рисунок 3.3 — Схема інтерфейсу взаємодії користувача з системою

Розроблений шаблон працює наступним чином:

- система виводить в консоль питання;

- система виводить в консоль варіанти відповідей з поясненням (зазвичай текст такий «1 — так, 0 — ні»);
- система виводить в консоль рядок «Ваша відповідь:»;
- користувач вводить необхідну відповідь;
- система виводить наступне питання.

Розроблена блок-схема буде використана в подальшому процесі реалізації як шаблон взаємодії користувача з системою.

### **Висновки до розділу 3**

В ході написання даного розділу спроектовано реалізацію моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.

Сформульовано основні функціональні вимоги до системи прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.

Запропоновано функціональну модель для забезпечення цих вимог в рамках реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.

Створено шаблон інтерфейсу взаємодії користувача з системою прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.

Проведено аналіз та проектування компонентного складу системи прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.

Усі отримані в ході роботи дані стануть основою для подальшого процесу реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.



## **Розділ 4**

### **Дослідження ефективності інформаційної технології для забезпечення підтримки екологічної рівноваги**

#### **4.1 Обґрунтування мови програмування для розробки реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж.**

Після отримання рішення про реалізацію розробленої магматичної моделі засобами технологій нейронних мереж питання стосовно вибору мови програмування, яка буде використовуватися для створення програмного продукту вирішилося на користь мови програмування Python, як основної мови для створення нейронних мереж в сучасному світі.

Python — це інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Його високорівневі вбудовані структури даних у поєднанні з динамічним типізацією та динамічним зв'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів. Простий, легкий у засвоєнні синтаксис Python підкреслює читабельність і, отже, знижує витрати на обслуговування програми. Python підтримує модулі та пакунки, що сприяє модульності програм і повторному використанню коду. Інтерпретатор Python і велика стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно поширюватися.

Часто програмісти використовують Python через підвищену продуктивність, яку він забезпечує. Оскільки етапу компіляції немає, цикл редагування-тестування-налагодження неймовірно швидкий. Налагоджувати програми Python легко: помилка або неправильне введення ніколи не призведе

до помилки сегментації. Натомість, коли інтерпретатор виявляє помилку, він створює виняток. Якщо програма не вловлює виняток, інтерпретатор друкує трасування стека. Налagodжувач рівня вихідного коду дозволяє перевіряти локальні та глобальні змінні, оцінювати довільні вирази, встановлювати точки зупини, переходити через код по рядку і так далі. Налagodжувач написаний на самому Python, що свідчить про інтроспективну силу Python. З іншого боку, часто найшвидший спосіб налагодити програму – це додати кілька операторів `print` до джерела: швидкий цикл редагування-тестування-налагодження робить цей простий підхід дуже ефективним.

Python часто порівнюють з іншими інтерпретованими мовами, такими як Java, JavaScript, Perl, Tcl або Smalltalk. Порівняння з C++, Common Lisp і Scheme також можуть бути просвітливими. На практиці вибір мови програмування часто диктується іншими реальними обмеженнями, такими як вартість, доступність, навчання, попередні інвестиції або навіть емоційна прихильність.

## Java

Зазвичай очікується, що програми на Python працюватимуть повільніше, ніж програми на Java, але вони також займають набагато менше часу на розробку. Програми на Python зазвичай в 3-5 разів коротші, ніж еквівалентні програми на Java. Цю різницю можна пояснити вбудованими високорівневими типами даних Python та його динамічним типом. Наприклад, програміст Python не витрачає часу на оголошення типів аргументів або змінних, а потужний поліморфний список і типи словників Python, для яких багата синтаксична підтримка вбудована прямо в мову, знаходять застосування майже в кожній програмі Python. Через введення під час виконання час виконання Python повинен працювати інтенсивніше, ніж Java. Наприклад, оцінюючи вираз `a+b`, він повинен спочатку перевірити об'єкти `a` і `b`, щоб з'ясувати їх тип, який невідомий під час компіляції. Потім він викликає відповідну операцію додавання, яка може бути перевантаженим методом, визначеним користувачем. Java, з іншого боку, може виконувати ефективне додавання цілого або з плаваючою комою, але

вимагає оголошення змінних для  $a$  і  $b$  і не допускає перевантаження оператора  $+$  для екземплярів визначених користувачем класів.

З цих причин Python набагато краще підходить як мова «клей», тоді як Java краще характеризувати як мову реалізації низького рівня. Насправді, ці два разом створюють чудову комбінацію. Компоненти можна розробляти на Java та об'єднувати для формування додатків на Python; Python також можна використовувати для створення прототипів компонентів, поки їхній дизайн не буде «загартований» у реалізації Java. Для підтримки такого типу розробки розробляється реалізація Python, написана на Java, яка дозволяє викликати код Python з Java і навпаки. У цій реалізації вихідний код Python перекладається в байт-код Java (за допомогою бібліотеки часу виконання для підтримки динамічної семантики Python).

### Javascript

«Об'єктна» підмножина Python приблизно еквівалентна JavaScript. Як і JavaScript (і на відміну від Java), Python підтримує стиль програмування, який використовує прості функції та змінні без участі в визначеннях класів. Однак для JavaScript це все, що є. Python, з іншого боку, підтримує написання набагато більших програм і кращого повторного використання коду завдяки справжньому об'єктно-орієнтованому стилю програмування, де класи та спадкування відіграють важливу роль.

### Perl

Python і Perl походять із схожого досвіду (скрипти Unix, які давно переросли) і мають багато схожих функцій, але мають іншу філософію. Perl наголошує на підтримці типових завдань, орієнтованих на програми, наприклад, вбудовані регулярні вирази, функції сканування файлів і створення звітів. Python наголошує на підтримці загальних методологій програмування, таких як розробка структури даних та об'єктно-орієнтоване програмування, і заохочує програмістів писати читабельний (і, отже, підтримуваний) код, надаючи

елегантну, але не надто загадкову нотацію. Як наслідок, Python наближається до Perl, але рідко перевершує його у своїй початковій області застосування; однак Python може застосовуватися далеко за межами ніші Perl.

## Tcl

Як і Python, Tcl можна використовувати як мову розширення програми, а також як окрему мову програмування. Однак Tcl, який традиційно зберігає всі дані у вигляді рядків, слабкий щодо структур даних і виконує типовий код набагато повільніше, ніж Python. Tcl також не має функцій, необхідних для написання великих програм, таких як модульні простори імен. Таким чином, хоча «типова» велика програма, що використовує Tcl, зазвичай містить розширення Tcl, написані на C або C++, які є специфічними для цієї програми, еквівалентну програму Python часто можна написати на «чистому Python». Звичайно, розробка на чистому Python набагато швидша, ніж писати й налагоджувати компонент C або C++. Було сказано, що єдиною корисною якістю Tcl є набір інструментів Tk. Python прийняв інтерфейс до Tk як стандартну бібліотеку компонентів графічного інтерфейсу.

Tcl 8.0 вирішує проблему швидкості, надаючи компілятор байт-коду з обмеженою підтримкою типів даних і додає простори імен. Однак це все ще набагато більш громіздка мова програмування.

## Smalltalk

Можливо, найбільша різниця між Python і Smalltalk полягає в більш «основному» синтаксисі Python, який дає йому перевагу в навчанні програміста. Як і Smalltalk, Python має динамічний введення та зв'язування, і все в Python є об'єктом. Однак Python відрізняє вбудовані типи об'єктів від визначених користувачем класів і наразі не дозволяє успадковувати вбудовані типи. Стандартна бібліотека типів даних колекції Smalltalk є більш досконалою, тоді як бібліотека Python має більше можливостей для роботи з Інтернетом та реальністю WWW, такими як електронна пошта, HTML і FTP.

Python має іншу філософію щодо середовища розробки та поширення коду. Якщо Smalltalk традиційно має монолітний «образ системи», який містить як середовище, так і програму користувача, Python зберігає як стандартні модулі, так і користувацькі модулі в окремих файлах, які можна легко перегрупувати або розповсюдити за межами системи. Одним із наслідків є те, що існує більше одного варіанту підключення графічного інтерфейсу користувача (GUI) до програми Python, оскільки графічний інтерфейс не вбудований в систему.

### C++

Майже все, що сказано для Java, також стосується C++, тільки більше: якщо код Python зазвичай в 3-5 разів коротший за еквівалентний код Java, він часто в 5-10 разів коротший за еквівалентний код C++.

### Common Lisp і Scheme

Ці мови близькі до Python за своєю динамічною семантикою, але дуже різні за своїм підходом до синтаксису. Слід зазначити, що Python має інтроспективні можливості, подібні до можливостей Lisp, а програми Python можуть створювати та виконувати фрагменти програми на льоту. Зазвичай властивості реального світу мають вирішальне значення: Common Lisp великий (у всіх сенсах), а світ Scheme фрагментований між багатьма несумісними версіями, де Python має єдину, безкоштовну, компактну реалізацію.

Виходячи з порівняння Python з відомими на цей час мовами програмування, основною мовою розробки даного програмного забезпечення було обрано саме Python.

## **4.2 Обґрунтування вибору програмної технології для розробки реалізації моделі прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж**

Керуючись інформацією, отриманої в розділі 4.1 було вирішено розробляти системуприйняття рішень по вибору методів переробки сміття на

базі технологій нейронних мереж з використанням мови програмування Python. Для прискорення і полегшення процесу буде використано середовище розробки PyCharm і бібліотеку scikit-learn.

PyCharm - це інтегроване середовище розробки (IDE), що використовується в комп'ютерному програмуванні, зокрема для мови Python. Він розроблений чеською компанією JetBrains [5]. Він забезпечує аналіз коду, графічний налагоджувач, інтегрований тестер одиниць, інтеграцію із системами контролю версій (VCSes), а також підтримує веб-розробку з Django, а також аналіз даних з Anaconda [6].

PyCharm є крос-платформним, з версіями для Windows, macOS та Linux. Видання Community випускається за ліцензією Apache [7], а також існує Professional Edition з додатковими функціями - випущена під власною ліцензією.

#### Основні переваги

- надання допомоги та аналіз, із заповненням коду, виділенням синтаксису та помилок, інтеграцією лінтерів та швидкими виправленнями
- Навігація проектом та кодом: спеціалізовані подання проектів, подання структури файлів та швидке переключення між файлами, класами, методами та звичками
- Рефакторинг Python: включає перейменування, метод вилучення, введення змінної, введення константи, підтягування, натискання та інші
- Підтримка веб-фреймворків: Django, web2py та Flask [лише для професійних версій] [8]
- Вбудований налагоджувач Python
- Комплексне модульне тестування з покриттям коду за рядками
- Розробка Google App Engine Python [лише для професійної версії]
- Інтеграція контролю версій: уніфікований користувальницький інтерфейс для Mercurial, Git, Subversion, Perforce та CVS зі списками змін та злиттям
- Підтримка наукових інструментів, таких як matplotlib, numpy та scipy [9]

Він конкурує в основному з низкою інших орієнтованих на Python середовищ IDE, включаючи PyDev від Eclipse та ширше зосереджену середовище розробки середовищ Komodo.

PyCharm надає API, щоб розробники могли писати власні плагіни для розширення функцій PyCharm. Кілька плагінів з інших середовищ IDE JetBrains також працюють з PyCharm. Існує більше 1000 плагінів, сумісних з PyCharm [10].

Бета-версія була випущена в липні 2010 року, а версія 1.0 з'явилася через 3 місяці. Версія 2.0 була випущена 13 грудня 2011 року, версія 3.0 - 24 вересня 2013 року, а версія 4.0 - 19 листопада 2014 року [11].

PyCharm Community Edition, відкрита версія PyCharm, стала доступною 22 жовтня 2013 р [7].

Scikit-learn (раніше scikits.learn і також відомий як sklearn) — це безкоштовна бібліотека машинного навчання для мови програмування Python.[3] Він містить різні алгоритми класифікації, регресії та кластеризації, включаючи машини опорних векторів, випадкові ліси, підвищення градієнта, k-середніх і DBSCAN, і розроблений для взаємодії з числовими та науковими бібліотеками PythonNumPy і SciPy. Scikit-learn — це фінансовий проект NumFOCUS.[4]

Проект scikit-learn розпочався як scikits.learn, проект GoogleSummerofCode французького вченого Девіда Курнапо. Його назва походить від уявлення про те, що це «SciKit» (SciPyToolkit), окремо розроблене та поширене стороннє розширення для SciPy.[5] Початкова кодова база пізніше була переписана іншими розробниками. У 2010 році Фабіан Педрегоза, Гаель Вароку, Олександр Грамфор і Вінсент Мішель, усі з Французького інституту досліджень комп'ютерних наук та автоматизації в Роккунку, Франція, взяли на себе керівництво проектом і зробили перший публічний випуск 1 лютого 2010 року.[6] ] З різних scikits, scikit-learn, а також scikit-image були описані як «доглянуті та популярні» в листопаді 2012 року[7].Scikit-learn є однією з найпопулярніших бібліотек машинного навчання на GitHub[8].

Scikit-learn в основному написаний на Python і широко використовує NumPy для високопродуктивної лінійної алгебри та операцій з масивами. Крім того, деякі основні алгоритми написані на Cython для підвищення продуктивності. Машини опорних векторів реалізовані за допомогою обгортки Cython навколо LIBSVM; логістичну регресію та лінійні опорні векторні машини за аналогічною обгорткою навколо LIBLINEAR. У таких випадках розширення цих методів за допомогою Python може бути неможливим.

Scikit-learn добре інтегрується з багатьма іншими бібліотеками Python, такими як Matplotlib і plotly для побудови графіків, NumPy для векторизації масивів, Pandas dataframes, SciPy та багатьма іншими.

### **4.3 Розробка системи прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж**

Розроблені модулі надають наступні функціональні можливості:

- інтерфейс взаємодії користувача з системою;
- обробка даних для передання на вхід нейронної мережі;
- розпаковка нейронної мережі по вибору методів переробки сміття;
- передача вхідних даних нейронній мережі;
- прийняття рішення по вибору методів переробки сміття;
- приведення отриманого результату вибору методів переробки сміття в формат, придатний для розуміння користувачем.

Продемонструємо життєвий цикл розробленого програмного продукту. При запуску проекту система отримує запит на отримання даних про відходи у користувача і реалізує його, передаючи інформацію у вигляді масиву до модулю роботи з нейронною мережею:

```
def get_predict():
```

```
print('Дайте відповіді на декілька питань, щоб визначити тип переробки
відходів.\n')
quest1 = float(input('Чи є на маркуванні знак повторного використання?
\n0 - ні \n1 - так\nВідповідь: '))
quest2 = float(input('Відходи горючі? \n0 - ні \n1 - так\nВідповідь: '))
quest3 = float(input('Відходи металічні? \n0 - ні \n1 - так\nВідповідь: '))
quest4 = float(input('Відходи органічні? \n0 - ні \n1 - так\nВідповідь: '))
data_arr = [quest1, quest2, quest3, quest4]
res = int(nh.get_predict(data_arr))
```

Початок роботи з програмною реалізацією зображено на рисунку 4.1.

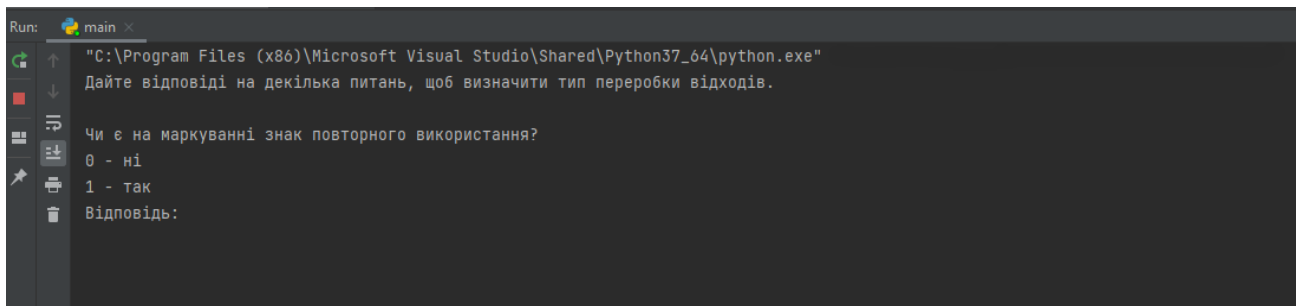


Рисунок 4.1 — Початок роботи з програмною реалізацією

Система не переходить до наступного модуля доки користувач не дасть відповіді на всі питання (рис. 4.2).

```

Run: main x
"C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe"
Дайте відповіді на декілька питань, щоб визначити тип переробки відходів.

Чи є на маркуванні знак повторного використання?
0 - ні
1 - так
Відповідь: 1
Відходи горючі?
0 - ні
1 - так
Відповідь: 1
Відходи металічні?
0 - ні
1 - так
Відповідь: 0
Відходи органічні?
0 - ні
1 - так
Відповідь: 1

```

Рисунок 4.2 — Ланцюг отримання необхідних даних від користувача

Після звернення модуля взаємодії з користувачем, модуль роботи з нейронною мережею перетворює масив у NumPy-масив, розпаковує модель, передає їх отримані дані та повертає результат у модуль взаємодії з користувачем:

```

def get_predict(arr):
    arr = np.array(arr).reshape((1,-1))
    with open("model.pkl", 'rb') as f:
        model_year = pickle.load(f)
    return model_year.predict(arr)

```

Після отримання результату модуль взаємодії з користувачем перетворює числовий еквівалент методу у назву методу і виводить її користувачеві:

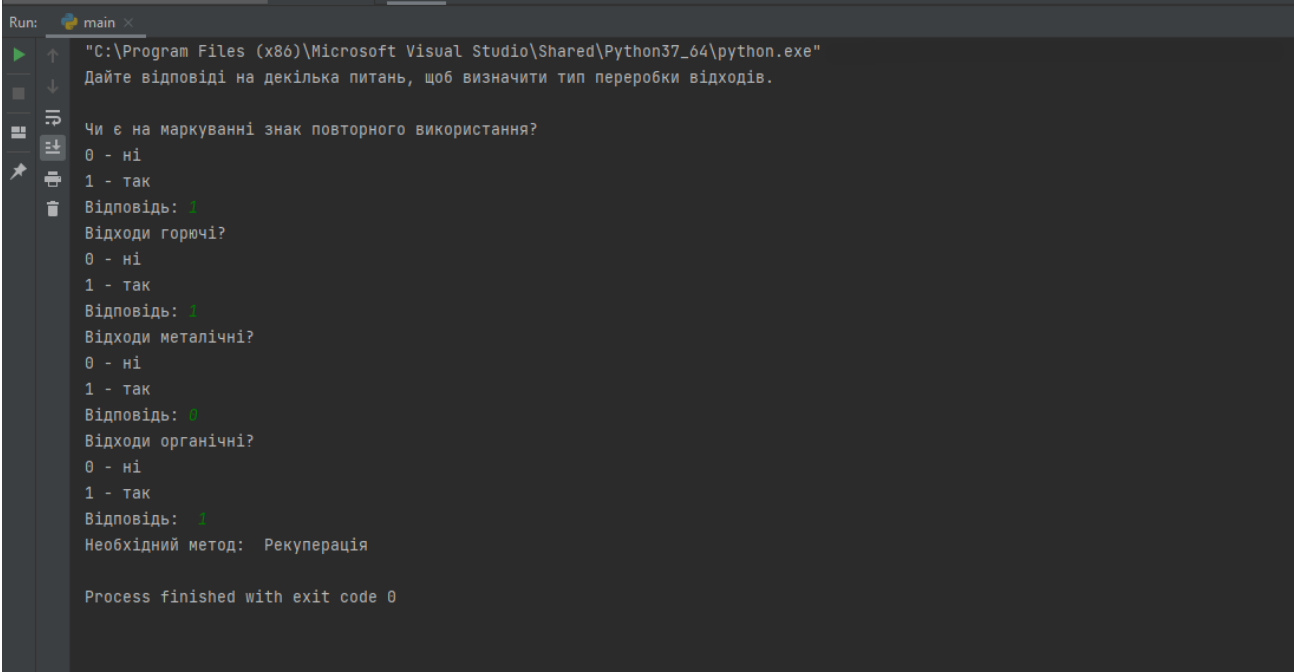
```

answ = ""
if res == 1:
    answ = 'Рекуперація'
elif res == 2:
    answ = 'Вторинна переробка'
elif res == 3:
    answ = 'Утилізація'

```

```
print('Необхідний метод: ', answ)
```

Вивід результату роботи показано на рисунку 4.3.



```
Run: main x
"C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe"
Дайте відповіді на декілька питань, щоб визначити тип переробки відходів.

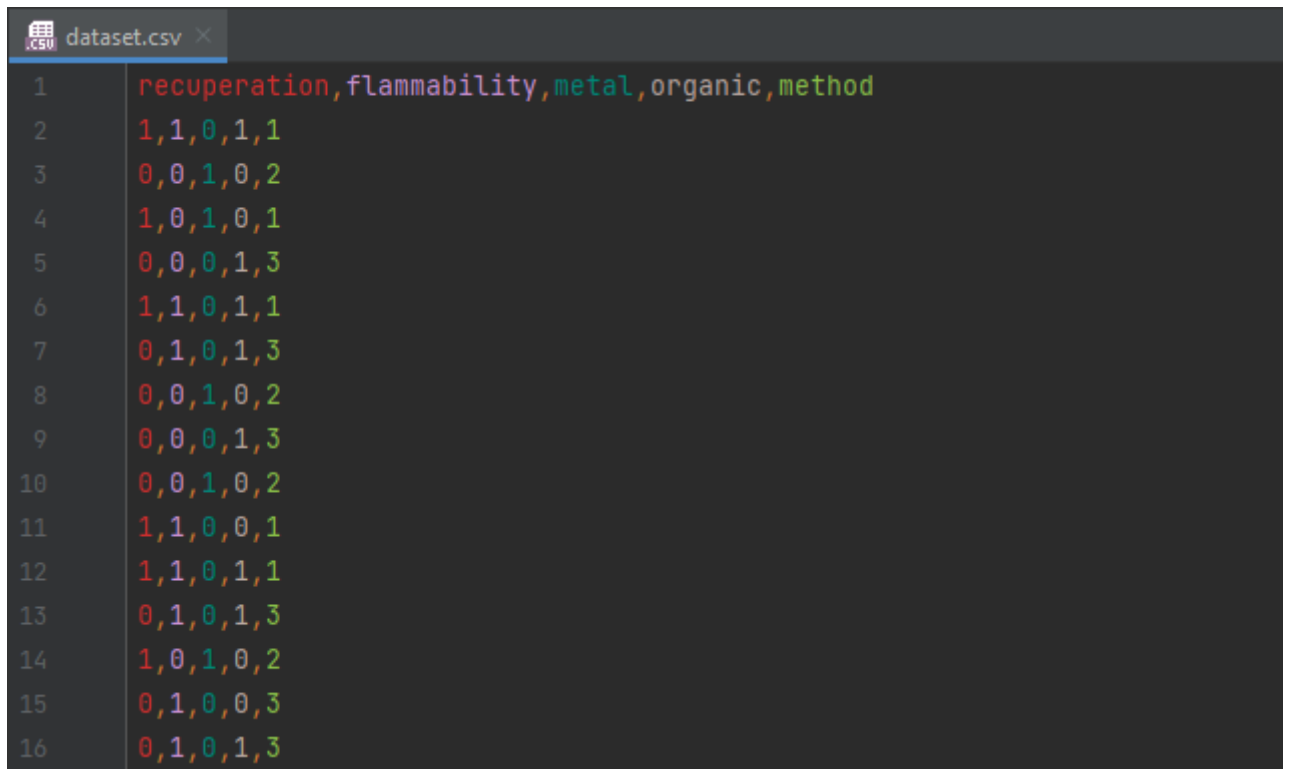
Чи є на маркуванні знак повторного використання?
0 - ні
1 - так
Відповідь: 1
Відходи горючі?
0 - ні
1 - так
Відповідь: 1
Відходи металічні?
0 - ні
1 - так
Відповідь: 0
Відходи органічні?
0 - ні
1 - так
Відповідь: 1
Необхідний метод: Рекуперация

Process finished with exit code 0
```

Рисунок 4.3 — Повний результат роботи програми

#### **4.4 Розробка модуля нейронної мережі системи прийняття рішень по вибору методів переробки сміття на базі технологій нейронних мереж**

Нейронна мережа представляє собою реалізацію `DecisionTreeClassifier`, який навчено на датасеті, що містить в собі дані стосовно вибору методу переробки відходів на базі їх характеристик. Частина навчального датасету зображено на рисунку 4.4.



	recuperation	flammability	metal	organic	method
1	1	1	0	1	1
2	0	0	1	0	2
3	1	0	1	0	1
4	0	0	0	1	3
5	1	1	0	1	1
6	0	1	0	1	3
7	0	0	1	0	2
8	0	0	0	1	3
9	0	0	1	0	2
10	1	1	0	0	1
11	1	1	0	1	1
12	0	1	0	1	3
13	1	0	1	0	2
14	0	1	0	0	3
15	0	1	0	1	3
16					

Рисунок 4.4 — Навчальний датасет

Навчання нейронної мережі відбувалося наступним чином:

```
df = pd.read_csv("dataset.csv")
x = df.iloc[:, :4]
y = df.iloc[:, 4:5]
model_year = tree.DecisionTreeClassifier(criterion="entropy")
model_year.fit(x.values, y.values)
model_year.score(x.values, y.values)
with open('model.pkl', 'wb') as f:
    pickle.dump(model_year, f)
```

#### Висновки до розділу 4

В ході написання четвертого розділу було описано особливості написання програмного коду інформаційної системи автоматизованої підтримки екологічної рівноваги у сфері переробки відходів. Було загострено увагу на

особливо важливих етапах розробки, приведено частини програмного коду, які описують дані моменти розробки.

Також було проаналізовано особливості функціоналу, виявлено основні функції та показано їх реалізацію.

Даний розділ несе в собі мету описати розроблену систему з точки зору програмного коду і структури програмного забезпечення, яке стало результатом даної роботи.

## Загальні висновки

Перший розділ роботи присвячено аналізу теоретичних аспектів предметної області і наукової розробленості питання розробки системи для збереження екологічної рівноваги у сфері переробки відходів.

Другий розділ роботи присвячений побудові математичної моделі забезпечення підтримки екологічної рівноваги.

Третій розділ роботи присвячено проектуванню і розробленню системи підтримки екологічної рівноваги в сфері переробки сміття засобами нейронних мереж.

В ході написання даного розділу сформовано функціональні вимоги до системи, створено функціональну схему системи підтримки екологічної рівноваги в сфері переробки.

Окрім цього в ході третього розділу проаналізовано і спроектовано компонентний склад системи і її інтерфейсну частину.

Четвертий розділ роботи присвячено вибору інструментальних засобів розробки системи підтримки екологічної рівноваги в сфері переробки та розробці окремих модулів даної системи.

Спираючись на отримані результати дослідження на базі створеної інформаційної системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів, отримано наступні результати:

- розроблена модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів продемонструвала ефективність за результатами дослідження;
- обрана оптимальна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Оцінюючи результати дослідження можна зробити висновок про виконання мети і завдання дослідження розробці оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів



## Перелік посилань

1. Villalba, G; Segarra, M; Fernández, A.I; Chimenos, J.M; Espiell, F (December 2002). "A proposal for quantifying the recyclability of materials". *Resources, Conservation and Recycling*. 37 (1): 39-53.doi:10.1016/S0921-3449(02)00056-3.
2. Lienig, Jens; Bruemmer, Hans (2017). "Recycling Requirements and Design for Environmental Compliance". *Fundamentals of Electronic Systems Design*. pp. 193–218. doi:10.1007/978-3-319-55840-0\_7. ISBN 978-3-319-55839-4.
3. European Commission (2014). "EU Waste Legislation". Archived from the original on 12 March 2014.
4. Geissdoerfer, Martin; Savaget, Paulo; Bocken, Nancy M.P.; Hultink, Erik Jan (1 February 2017). "The Circular Economy – A new sustainability paradigm?" (PDF). *Journal of Cleaner Production*. 143: 757–768. doi:10.1016/j.jclepro.2016.12.048. S2CID 157449142.
5. "7 Things You Didn't Know About Plastic (and Recycling)" *National Geographic*. Retrieved 26 July 2019.
6. Black Dog Publishing (2006). *Recycle : a source book*. London, UK: Black Dog Publishing. ISBN 978-1-904772-36-1.
7. "The truth about recycling". *The Economist*. 7 June 2007.
8. Cleveland, Cutler J.; Morris, Christopher G. (15 November 2013). *Handbook of Energy: Chronologies, Top Ten Lists, and Word Clouds*. Elsevier. p. 461. ISBN 978-0-12-417019-3.
9. Dadd-Redalia, Debra (1 January 1994). *Sustaining the earth: choosing consumer products that are safe for you, your family, and the earth*. New York: Hearst Books. p. 103. ISBN 978-0-688-12335-2. OCLC 29702410.
10. Nongpluh, Yoofisaca Syngkon. (2013). *Know all about : reduce, reuse, recycle*. Noronha, Guy C., Energy and Resources Institute. New Delhi. ISBN 978-1-4619-4003-6. OCLC 858862026.

11. Carl A. Zimring (2005). Cash for Your Trash: Scrap Recycling in America. New Brunswick, NJ: Rutgers University Press. ISBN 978-0-8135-4694-0.
12. Rethinking economic incentives for separate collection. Zero Waste Europe & ReLoop Platform, 2017
13. "Report: "On the Making of Silk Purses from Sows' Ears," 1921: Exhibits: Institute Archives & Special Collections: MIT". mit.edu. Archived from the original on 3 June 2016. Retrieved 7 July 2016.
14. Public Broadcasting System (2007). "The War Episode 2: Rationing and Recycling". Retrieved 7 July 2016.
15. Out of the Garbage-Pail into the Fire: fuel bricks now added to the list of things salvaged by science from the nation's waste, Popular Science monthly, February 1919, page 50-51, Scanned by Google Books: <https://books.google.com/books?id=7igDAAAAMBAAJ&pg=PA50>
16. "Recycling through the ages: 1970s". Plastic Expert. Plastic Expert. 30 July 2014. Retrieved 7 March 2015.
17. "The price of virtue". The Economist. 7 June 2007.
18. "CRC History – Computer Recycling Center". www.crc.org. Retrieved 29 July 2015.
19. "About us – Swico Recycling". www.swicorecycling.ch. Retrieved 29 July 2015.
20. "Where does e-waste end up?". www.greenpeace.org/. Greenpeace. 24 February 2009. Retrieved 29 July 2015.
21. Програмне забезпечення ТОВ "Фірма Інтеграл" СЕРІЇ "ЕКОЛОГ" URL: <https://integral.ru/>.
22. EcoReport URL: <http://ecoreport.ru>.
23. Hydromantis Environmental Software Solutions, Inc URL: <https://www.hydromantis.com/>
24. Пакет ТЕПЛО URL: <http://www.potok.ru/>.
25. "Bulgaria opens largest WEEE recycling factory in Eastern Europe". www.ask-eu.com. WtERT Germany GmbH. 12 July 2010. Retrieved 29 July 2015.

26. Moses, Asher (19 November 2008). "New plant tackles our electronic leftovers – BizTech –Technology –smh.com.au". [www.smh.com.au](http://www.smh.com.au). Retrieved 29 July 2015.
27. European Commission, Recycling Archived 3 February 2014 at the Wayback Machine.
28. Recycling rates in Europe, European Environment Agency.
29. Recycling of municipal waste, European Environment Agency, 30 November 2017. (Dutch)
30. Germany's recycling rate continues to lead Europe, Resource Recycling, 7 February 2017. (Dutch)
31. Hook, Leslie; Reed, John (24 October 2018). "Why the world's recycling system stopped working". *Financial Times*. Archived from the original on 25 October 2018. Retrieved 25 October 2018.
32. DEFRA (2013). "Quality Action Plan Proposals to Promote High Quality Recycling of Dry Recyclates" (PDF).
33. "How to Recycle Tin or Steel Cans" Earth911.com. Retrieved 26 July 2019.
34. The Scottish Government (2012) (5 October 2012). "Recyclate Quality Action Plan – Consultation Paper".
35. The Highland Council (2013). "Report by Director of Transport, Environmental and Community Services" (PDF). Archived from the original (PDF) on 7 April 2014.
36. Waldrop, M. Mitchell (1 October 2020). "One bin future: How mixing trash and recycling can work". *Knowable Magazine*. doi:10.1146/knowable-092920-3. S2CID 224860591.
37. "The State of Multi-Tenant Recycling in Oregon" (PDF). April 2018.
38. Singer, Paul (21 April 2017). "Recycling market in a heap of trouble". *USA Today*. Melbourne, Florida. pp. 1B, 2B. Retrieved 21 April 2017.
39. Дудар О.В. Інформаційна система для забезпечення підтримки екологічної рівноваги/ Дудар О. В., Міхалевський В. Ц., Скрипник Т.К. // Збірник наукових праць за матеріалами XIII всеукраїнської науково-практичної

конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». – Хмельницький, 2021. – С. 321-323.

# ДОДАТКИ

## Додаток А

### Лістинг програмного коду

```
import numpy as np
import pandas as pd
from sklearn import tree
import pickle

df = pd.read_csv("dataset.csv")
x = df.iloc[:, :4]
y = df.iloc[:, 4:5]

model_year = tree.DecisionTreeClassifier(criterion="entropy")
model_year.fit(x.values, y.values)
model_year.score(x.values, y.values)

with open('model.pkl', 'wb') as f:
    pickle.dump(model_year, f)

import network as nh

def get_predict():
    print('Дайте відповіді на декілька питань, щоб визначити тип переробки
відходів.\n')
```

```
quest1 = float(input('Чи є на маркуванні знак повторного використання?
\n0 - ні \n1 - так\nВідповідь: '))
quest2 = float(input('Відходи горючі? \n0 - ні \n1 - так\nВідповідь: '))
quest3 = float(input('Відходи металічні? \n0 - ні \n1 - так\nВідповідь: '))
quest4 = float(input('Відходи органічні? \n0 - ні \n1 - так\nВідповідь: '))
data_arr = [quest1, quest2, quest3, quest4]
res = int(nh.get_predict(data_arr))
answ = ""
if res == 1:
    answ = 'Рекуперація'
elif res == 2:
    answ = 'Вторинна переробка'
elif res == 3:
    answ = 'Утилізація'
print('Необхідний метод: ', answ)
get_predict()

import numpy as np
import pickle

def get_predict(arr):
    arr = np.array(arr).reshape((1,-1))
    with open("model.pkl", 'rb') as f:
        model_year = pickle.load(f)
    return model_year.predict(arr)
```

## **Додаток Б**

### **Ксерокопії наукових публікацій, виконаних при роботі над кваліфікаційною роботою магістра**

*(ксерокопії титульної сторінки, сторінки змісту та всіх сторінок із  
публікацією)*

#### **Перелік наукових публікацій**

1. Дудар О.В. Інформаційна система для забезпечення підтримки екологічної рівноваги/ Дудар О. В., Міхалевський В. Ц., Скрипник Т.К. // Збірник наукових праць за матеріалами XIII всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». – Хмельницький, 2021. – С. 321-323.



Міністерство освіти і науки України  
Хмельницький національний університет



**ЗБІРНИК НАУКОВИХ ПРАЦЬ**  
за матеріалами XIII Всеукраїнської науково-практичної конференції  
«Актуальні проблеми комп'ютерних наук АПКН-2021»

*15-16 жовтня 2021*

Хмельницький 2021

УДК 004:37:001:62

Збірник наукових праць за матеріалами XIII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021». Хмельницький – 2021. – 413с.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів.

Участь у конференції та складові всіх її етапів (розгляд праць, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: [apkt.khnu@gmail.com](mailto:apkt.khnu@gmail.com)

© 2021 Хмельницький національний університет

© 2021 Кафедра комп'ютерних наук ХНУ

<b>Галкіна Р. І., Багрій Р. О., Скрипник Т. К.</b> Застосування адаптивного підходу для реалізації системи опитувань та тестувань.....	306
<b>Гринь С. С., Пивовар О. С., Таранчук А. А.</b> Забезпечення прихованості дії та криптографічного захисту аналогових сигналів в хаотичній системі зв'язку.....	309
<b>Данчук С. В., Багрій Р. О.</b> Технологія автоматизованого отримання даних з веб-ресурсів для бізнес-аналітики.....	312
<b>Длугунович Н. А.</b> Інформаційна технологія фінансового моделювання для розвитку малого підприємництва.....	316
<b>Дрозд А. І., Форкун Ю. В.</b> Метод розподілу обчислювальних ресурсів для обробки розподілених потоків даних.....	319
<b>Дудар О. В., Міхалевський В. Ц., Скрипник Т. К.</b> Інформаційна система для забезпечення підтримки екологічної рівноваги.....	321
<b>Єфімчук А. С., Скрипник Т. К., Мазурець О. В., Молчанова М. О.</b> Автоматизований розподіл процесів при управлінні ІТ-проєктами в складних критично-безпекових умовах.....	324
<b>Житкевич В. В., Медведчук В. Ю.</b> Метод відновлення пошкоджених растрових зображень.....	332
<b>Заровний В. І., Скрипник Т. К.</b> Методи шифрованої передачі даних між хмарними підпросторами.....	335
<b>Кудрявцев В. В., Форкун Ю. В.</b> Аналіз та застосування методів оптимізації швидкодії та відмовостійкості програмних продуктів.....	338
<b>Курдибаха А. В., Мазурець О. В., Собко О. В., Молчанова М. О.</b> Інформаційна технологія оцінювання діяльності сімейного лікаря за даними прийомів.....	340
<b>Лаврентій А. А., Петровський С. С.</b> Метод оцінювання наповненості дистанційних курсів предметів у школі.....	349
<b>Левченко Т. В., Блажук В. Д., Молчанова М. О., Собко О. В.</b> Метод оптимізації транспортних перевезень засобами біологічної метаевристики.....	352

УДК 000.4

Дудар О. В., Міхалевський В. Ц., Скрипник Т. К.

*Хмельницький національний університет***ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ  
ЕКОЛОГІЧНОЇ РІВНОВАГИ**

*Проведено дослідження на тему автоматизації підтримки екологічної рівноваги, за рахунок розроблення інформаційної системи, націленої на автоматизацію і спрощення процесу очищення місць особливого забруднення з метою подальшої переробки відходів.*

*A study was conducted on the topic of automation of ecological balance maintenance, due to the development of an information system aimed at automation and simplification of the process of cleaning places of special pollution with the aim of further processing of waste.*

В сучасному світі питання підтримки екологічної рівноваги постає надзвичайно гостро на фоні різноманітних екологічних катастроф та моніторингу стану забруднення оточуючого середовища.

По прогнозам екологів через декілька століть придатність планети для життя впаде на 40-60%, при втриманні сучасного відношення до екології та показників забруднення оточення.

Одна з найважливіших частин процесу підтримки екологічної рівноваги — переробка відходів. Переробка - це процес перетворення відходів у нові матеріали та об'єкти. У цю концепцію часто входить відновлення енергії з відходів. Переробка матеріалу залежить від його здатності знову набувати властивостей, які він мав у вихідному стані [1]. Це альтернатива "звичайному" видаленню відходів, що дозволяє заощадити матеріал та зменшити викиди парникових газів. Він також може запобігти відходам потенційно корисних матеріалів та зменшити споживання свіжої сировини, зменшивши споживання енергії, забруднення повітря (від спалювання) та забруднення води (зі звалищ).

Переробка є ключовим компонентом сучасного скорочення відходів і є третім компонентом ієрархії відходів "Скорочення, повторне використання та переробка" [2, 3]. Він сприяє екологічній стійкості, вилучаючи сировинні ресурси та перенаправляючи викиди відходів в економічну систему [4].

Найпростішим з точки зору впровадження є автоматизація процесу підтримки екологічної рівноваги шляхом створення інформаційної системи, яка агрегує конкретні проблеми, наприклад засмічені місця в світі, і пропонує

волонтерам, які користуються ресурсом, провести зачистку цього місця і передати зібрані відходи до будь-якого центру переробки сміття.

Інформаційна система (ІС) - це формальна, соціально-технічна, організаційна система, призначена для збору, обробки, зберігання та розповсюдження інформації. У соціотехнічній перспективі інформаційні системи складаються з чотирьох компонентів: завдання, людей, структури (або ролей) та технології. Інформаційні системи можна визначити як інтеграцію компонентів для збору, зберігання та обробки даних, дані яких використовуються для надання інформації, сприяння знанням, а також цифрових продуктів.

В даному випадку інформаційна система представлена веб-сайтом. Такий вибір технологій зумовлений автономністю сайту, яка є важливим фактором проектування, виходячи з тематики ресурсу.

Створювана інформаційна система базується на моделі MVC, що дозволяє покращити швидкість роботи ресурсу і позбавити користувачів від необхідності завантажувати на власний девайс велику кількість графічного матеріалу та внутрішніх файлів ресурсу.

Схема роботи моделі показана на рисунку 1.

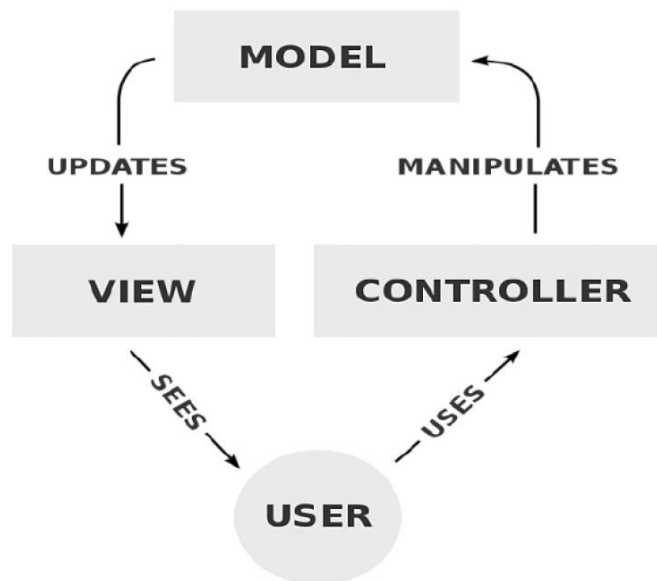


Рисунок 1 – Діаграма роботи MVC

Новизна розроблюваної системи полягає в:

- простоті взаємодії між користувачами та фактична відсутність адміністрації, оскільки завдання (далі «тікети») публікуються безпосередньо користувачами і виконуються іншими користувачами. Адміністрація ресурсу відповідає лише за фільтрацію і достовірність наданої інформації;

- принципово новій ідеї створення ресурсу, який покликаний агрегувати екологічні проблеми з метою їх подальшого усунення;
- наявність звітів про усунення різного роду забруднень, як доказу очищення опублікованого місця і мотивації для інших користувачів ресурсу.

Виходячи з усього вищесказаного, важко недооцінити актуальність і, навіть, необхідність подібного роду ресурсу і його наукову і технічну новизну в сучасному соціумі.

#### **Перелік посилань**

1. Villalba, G; Segarra, M; Fernández, A.I; Chimenos, J.M; Espiell, F (December 2002). "A proposal for quantifying the recyclability of materials".
2. Lienig, Jens; Bruemmer, Hans (2017). "Recycling Requirements and Design for Environmental Compliance". *Fundamentals of Electronic Systems Design*.
3. European Commission (2014). "EU Waste Legislation"
4. Geissdoerfer, Martin; Savaget, Paulo; Bocken, Nancy M.P.; Hultink, Erik Jan (1 February 2017). "The Circular Economy – A new sustainability paradigm?"

## Додаток В

### Презентаційні матеріали

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

## Забезпечення екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю

ВИКОНАВ:

СТУДЕНТ 2 КУРСУ, ГРУПА КНМ-20-1

ДУДАР ОЛЕКСАНДР ВІТАЛІЙОВИЧ

КЕРІВНИК:

К.Ф.-М.Н., ДОЦЕНТ КАФЕДРИ КН

МІХАЛЕВСЬКИЙ ВІТАЛІЙ ЦЕЗАРІЙОВИЧ

### АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

**Актуальність дослідження** зумовлена стрімким розвитком використання засобів автоматизації, таких як електронні системи обробки заявок в тій чи іншій тематиці, автоматизацією усіх процесів життєдіяльності людини.

Процес пошуку виконавця для виконання заявки на прибирання відходів в громадських місцях займає багато часу і потребує немало зусиль. Автоматизація даного процесу звільняє велику кількість часу і дозволяє підвищити кількість заявок, що позитивно вплине на основну діяльність, збереження екологічної рівноваги у сфері переробки відходів.

Тому система автоматизації забезпечення екологічної рівноваги у сфері переробки відходів є не тільки рекомендаційною до впровадження, а й абсолютною необхідністю, що суттєво оптимізує процес збору та переробки відходів.

Але існуючи на даний час системи автоматизації забезпечення екологічної рівноваги у сфері переробки відходів не використовують оптимізаційної моделі забезпечення підтримки екологічної рівноваги.

Цю проблему можна розв'язати за рахунок розробки оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Для розв'язання визначеної проблеми нами буде розроблена оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

## МЕТА РОБОТИ

**Мета кваліфікаційної роботи магістра** - теоретично обґрунтувати, розробити та апробувати систему для забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю.

**Об'єктом дослідження** є технологія забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю.

**Предметом дослідження** є система для збору та переробки усіх сфер діяльності.

Відповідно до предмету та мети дослідження визначено наступні **задачі**:

- Аналіз теоретичних підходів та стан наукових досліджень систем екологічної рівноваги у сфері переробки відходів.
- Розробка моделі системи для забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю.
- З'ясування особливостей використання програмно-апаратних засобів системи для забезпечення підтримки екологічної рівноваги у сфері переробки відходів.
- Наукове обґрунтування, розробка та експериментальна перевірка технології розробки системи для забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

## НАУКОВА НОВИЗНА ОДЕРЖАНИХ РЕЗУЛЬТАТІВ

В результаті проведеної роботи були отримані такі результати:

- набули подальшого розвитку існуючі моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів;
- досліджено практичну ефективність моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

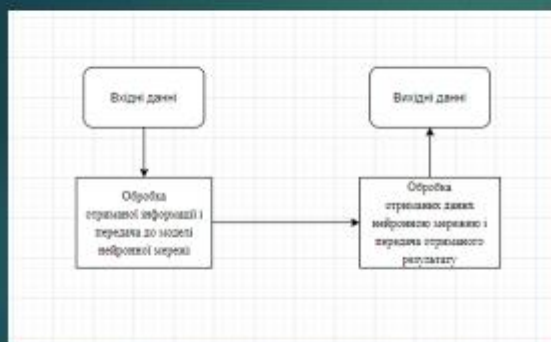
## Практичне значення одержаних результатів

На основі розробленої моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів для підтвердження результатів дослідження була створена інформаційна система забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Спираючись на отримані результати дослідження на базі створеної інформаційної системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів, отримано наступні результати:

- розроблена модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів продемонструвала;
- обрана оптимальна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

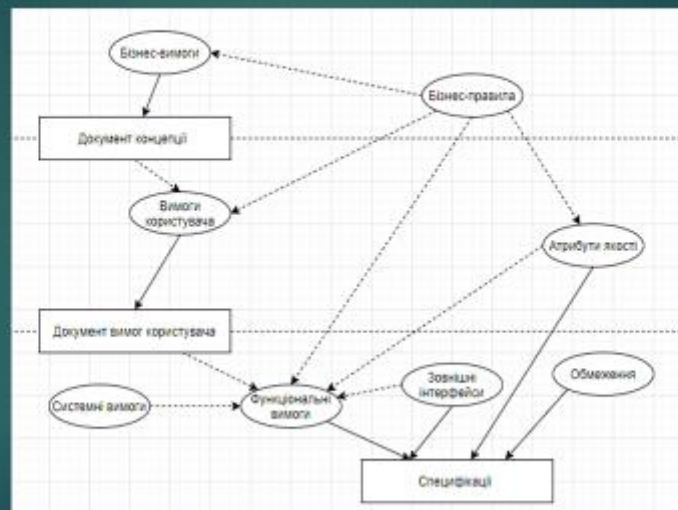
## СХЕМА РОБОТИ МОДЕЛІ ОПТИМІЗАЦІЇ ПЕРЕРОБКИ ВІДХОДІВ



Система повинна складатися з таких кроків:

1. Вхідна інформація: відповіді користувача на питання стосовно характеристик відходів.
2. Обробка отриманої інформації і передача до моделі нейронної мережі.
3. Обробка отриманих даних нейронною мережею і передача отриманого результату.
4. Вихідні дані: метод переробки відходів.

## ВЗАЄМОЗВ'ЯЗКИ ДЕКІЛЬКОХ ТИПІВ ІНФОРМАЦІЇ ДЛЯ ВИМОГ



## ЗАГАЛЬНІ ВИСНОВКИ

Перший розділ роботи присвячено аналізу теоретичних аспектів предметної області і науковій розробленості питання розробки системи для збереження екологічної рівноваги у сфері переробки відходів.

Другий розділ роботи присвячений побудові математичної моделі забезпечення підтримки екологічної рівноваги.

Третій розділ роботи присвячено проектуванню і розробленню системи підтримки екологічної рівноваги в сфері переробки сміття засобами нейронних мереж.

В ході написання даного розділу сформовано функціональні вимоги до системи, створено функціональну схему системи підтримки екологічної рівноваги в сфері переробки.

Окрім цього в ході третього розділу проаналізовано і спроектовано компонентний склад системи і її інтерфейсну частину.

Четвертий розділ роботи присвячено вибору інструментальних засобів розробки системи підтримки екологічної рівноваги в сфері переробки та розробці окремих модулів даної системи.

Спираючись на отримані результати дослідження на базі створеної інформаційної системи забезпечення підтримки екологічної рівноваги у сфері переробки відходів, отримано наступні результати:

Розроблена модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів продемонструвала:

Обрана оптимальна модель забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Оцінюючи результати дослідження можна зробити висновок про виконання мети і завдання дослідження розробці оптимізаційної моделі забезпечення підтримки екологічної рівноваги у сфері переробки відходів.



ДЯКУЮ ЗА УВАГУ!

17.12.2021, 11:59

result\_2598472634457157160.html

Fri Dec 17 11:11:34 EET 2021, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

## Anti-Plagiarism v-15.257

**Максимальное совпадение с одним документом 3.0%**
**Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 9%**

ID: 99849 Название: Забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю Добавлено в БД: 2021-12-17 Авторы: О.В. Дудар Руководители: В.Ц. Міхалевський Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	106997	816	6310 (6%)	61 (7%)

### Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:  
Кафедра КН

ID перевірки:  
1009704319

Дата перевірки:  
17.12.2021 11:48:08 EET

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
17.12.2021 11:51:28 EET

ID користувача:  
100005671

Назва документа: КРМ\_Дудар\_Lite

Кількість сторінок: 82 Кількість слів: 15622 Кількість символів: 125019 Розмір файлу: 1.25 MB ID файлу: 1009703182

## 6.77% Схожість

Найбільша схожість: 1.85% з джерелом з Бібліотеки (ID файлу: 1009695153)

5.2% Джерела з Інтернету	65	Сторінка 84
3.48% Джерела з Бібліотеки	92	Сторінка 85

## 0.23% Цитат

Цитати	8	Сторінка 86
Посилання	1	Сторінка 86

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	4
------------------	---



РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА ДО ЗАХИСТУ ЗА  
РЕЗУЛЬТАТАМИ АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю

Автор: Дудар О.В., група КНм-20-1

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.ф-м.н., доц. кафедри КН Міхалевський В.Ц.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) за програмою Anti-Plagiarism виявлені 3,0% запозичень, що підтверджує авторство дослідження.

2) За програмою UNICHECK виявлені 6,77% запозичень є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 3,0% і 6,77% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи \_\_\_\_\_

Гарант ОП \_\_\_\_\_

Завідувач кафедри КН \_\_\_\_\_

В. Ц. Міхалевський

Р. О. Багрій

О. В. Бармак



**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МОН УКРАЇНИ**



кафедра комп'ютерних наук

**ВІДГУК ОПОНЕНТА**

**на кваліфікаційну роботу магістра**

*гр. КНМ-20-1 Дударя Олександра Віталійовича за темою: Забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю*

**1. Актуальність обраної теми**

В кваліфікаційній роботі магістра було розроблено метод забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю. Тема роботи є актуальною на даний час, оскільки перед суспільством постали серйозні екологічні проблеми, викликані неконтрольованим забрудненням зовнішнього середовища, відсутністю екологічної культури щодо дотримання основних правил і циклів виробничих та суспільних процесів, неможливістю забезпечити вирішення завдань переробки відходів традиційними способами. Актуальність детально обґрунтована дослідженнями процесів накопичення, сортування та переробки відходів, методів прийняття рішень щодо забезпечення підтримки екологічної рівноваги та аналізом ряду програмних продуктів, що допомагають вирішувати задачу оптимізації проведення заходів з переробки відходів.

**2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт**

Предметна область процесу забезпечення підтримки екологічної рівноваги дозволяє формалізацію, систематизацію, збір та аналіз інформації для автоматизованого визначення та супроводження процесів накопичення, сортування та переробки відходів за множиною наявних параметрів. Розроблена інформаційна система описує, аналізує та оптимізує архітектурні рішення для автоматизованого забезпечення підтримки екологічної рівноваги у сфері переробки відходів.

Кваліфікаційна робота відповідає вимогам стандарту до роботи магістра за спеціальністю 122 Комп'ютерні науки.

**3. Повнота розкриття мети та завдань дослідження**

Завдання дослідження в достатній мірі розкривають мету роботи. Проаналізовано теоретичні підходи та стан наукових досліджень систем екологічної рівноваги у сфері переробки відходів та розроблено модель системи для супроводження процесів накопичення, сортування і переробки відходів за множиною наявних параметрів, що дозволяє формувати експертний висновок з поясненням заходів по забезпеченню екологічної рівноваги у сфері переробки відходів.

**4. Наявність наукової новизни**

Наукова новизна підтверджена розробкою оптимізаційної моделі системи для автоматизованого супроводження процесів накопичення, сортування і переробки відходів за множиною наявних параметрів, що дозволяє формувати експертний висновок з поясненням заходів по забезпеченню екологічної рівноваги у сфері переробки відходів. При цьому

запропоновано метод агрегації екологічних проблем для розробки та функціонування системи екологічної рівноваги.

#### **5. Зміст кожного розділу роботи**

В першому розділі проаналізована предметна область та сформована постановка задачі на розробку системи інформаційного забезпечення для автоматизації процесу підтримки екологічної рівноваги.

В другому розділі проаналізовано існуючі технології, зроблено загальний опис інформаційної технології забезпечення підтримки екологічної рівноваги та запропоновано і побудовано оптимізаційну модель системи для супроводження процесів накопичення, сортування і переробки відходів за множиною наявних параметрів, що дозволяє формувати експертний висновок з поясненням заходів по забезпеченню екологічної рівноваги у сфері переробки відходів.

В третьому розділі визначено комбінації засобів розробки інформаційної системи, спроектовано модель прийняття рішень по вибору методів переробки сміття на базі технології нейронних мереж та запропоновано схему взаємодії функціональних модулів інформаційної системи.

В четвертому розділі протестовано і досліджено ефективність інформаційної технології автоматизованої підтримки екологічної рівноваги у сфері переробки відходів та зроблена оцінка отриманих результатів.

#### **6. Ступінь розкриття теми роботи**

Тема роботи розкрита в достатній мірі. Достовірність результатів підтверджена процесом тестування на основі вхідних параметрів для автоматизації підтримки екологічної рівноваги, збору, сортування та переробки відходів.

#### **7. Якість оформлення кваліфікаційної роботи**

Пояснювальна записка кваліфікаційної роботи магістра оформлена відповідно до норм. Мовних, граматичних, синтаксичних помилок не виявлено.

#### **8. Недоліки кваліфікаційної роботи**

Явних недоліків в роботі не виявлено. Проте деякі аспекти, зокрема процеси збору і сортування відходів, розглянуті поверхово. Оптимізаційна модель вимагає детальнішого дослідження. Можна було б провести узагальнення роботи системи шляхом її демонстрації на різних етапах функціонування та спроектувати перспективу розвитку подібних систем. Не розроблено загальних підходів та рекомендацій щодо впровадження, супроводження та ефективного функціонування систем з підтримки екологічної рівноваги для різних видів виробничої діяльності.

#### **9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота.**

Рекомендую допустити кваліфікаційну роботу до захисту.

Робота заслуговує на оцінку « *задовільно* ».

Опонент



д.т.н., проф. Мартинюк В.В.



## ВІДГУК НАУКОВОГО КЕРІВНИКА

### на кваліфікаційну роботу магістра

*гр. КНМ-20-1 Дударя Олександра Віталійовича за темою: Забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю*

#### 1. Актуальність теми

Актуальність теми обґрунтована в достатній мірі: на сьогоднішній день існує ряд рішень, що допомагають вирішувати задачу забезпечення екологічної рівноваги, проте вони мають ряд недоліків. Враховуючи те, що багато процесів з обробки відходів мають нерегульований та стихійний характер, можливі помилки організації та прийняття рішень та з таких даних може бути визначений неповний перелік заходів, розробка методів і процесів збору, класифікації, обліку і переробки відходів на поточному етапі є актуальною.

#### 2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

В кваліфікаційній роботі магістра було розроблено метод забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю. Досліджено процеси збору, обліку, сортування та переробки відходів, методи прийняття рішень щодо організаційних заходів та проведено аналіз ряду програмних продуктів для забезпечення процесу переробки відходів.

Тема кваліфікаційної роботи магістра відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи магістра: 1) Виконавцем сплановано і реалізовано процес розробки комп'ютерної систем та програмного забезпечення, проведено тестування та порівняльний аналіз. 2) Виконавець обрав інформаційне середовище розробки та дослідження, що дозволило знайти правильне і ефективне рішення, а запропонований метод дозволяє обрати найоптимальніший алгоритм збору і переробки відходів.

Проаналізовано, оцінено та порівняно різні технології процесів для встановлення пріоритетів у відповідності з критеріями продуктивності та якості, що визначені завданням. Розроблено інформаційну систему автоматизованого забезпечення організаційно-виробничих процесів переробки відходів. Проведено функціональне та прикладне дослідження ефективності розроблених засобів, у тому числі для тестового набору стандартної множини значень системи.

#### 3. Професійні та особистісні якості магістранта

Магістрант володіє в достатній мірі професійними якостями дослідника: 1) Має здатність збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до комп'ютерної системи, що розробляється. 2) Має здатність формалізувати предметну область проєкту у вигляді відповідної інформаційної моделі.

Серед особистісних якостей магістранта слід виділити відповідальність, працездатність, бажання навчатися.

#### 4. Ступінь самостійності під час виконання кваліфікаційної роботи

Студент більшу частину роботи виконав самостійно. Особисто магістрантом досліджено предметну область, проведено порівняльний аналіз переваг та недоліків існуючих програмних засобів для забезпечення екологічної рівноваги; підібрано алгоритм для опрацювання даних; розроблено нову інформаційну систему автоматизованого

забезпечення збору та переробки відходів, проведено функціональне та прикладне дослідження ефективності розроблених засобів, у тому числі для тестового аналізу га стандартній множині значень системи.

#### **5. Наукова новизна та оригінальність запропонованих підходів**

У виконаній роботі наукова новизна присутня в достатній мірі. Інноваційний підхід проявлено в розробці методу автоматизованого забезпечення екологічної рівноваги за оптимізаційною моделлю, що дозволяє за множиною наявних атрибутів, параметрами наявних атрибутів та характером відходів автоматизовано організувати процес їх переробки.

Результати дослідження доповідались на 1-й конференції та оприлюднені в 1-х тезах.

#### **6. Ступінь оволодіння методами дослідження**

Магістрант в достатній мірі оволодів методами дослідження, які були використані у роботі: метод аналізу даних і теорії множин, методологія проектування інформаційних систем і об'єктно-орієнтований підхід.

#### **7. Повнота та якість розкриття теми роботи**

Тема роботи розкрита достатньо в рамках поставлених завдань: спроектовано структуру і здійснено прикладну програмну розробку інформаційної системи автоматизованого забезпечення підтримки екологічної рівноваги у сфері переробки відходів за оптимізаційною моделлю. Інформаційна система включає в себе базу даних, базу знань і п'ять функціональних модулів, що мають різне призначення: модуль роботи користувача з експертними даними, модуль роботи користувача з каталоговими даними, модуль взаємодії користувача з оперативними даними, модуль формування множини заходів з переробки відходів та модуль експертного пояснення прийнятого рішення і виведення користувачу результатів роботи.

#### **8. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу**

Магістрант матеріал виклав логічно, послідовно, аргументовано. Наводилися наявні розробки, ставилося завдання та послідовно розв'язувалося. Для аргументації отриманих рішень проводилося теоретичне обґрунтування та порівняльний аналіз експериментів.

Літературна та граматична якість матеріалу на достатньому рівні.

#### **9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин**

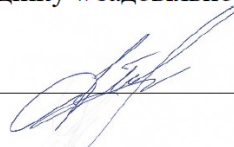
Результати кваліфікаційної роботи магістра можуть застосовуватися на практиці після налаштувань під конкретно запропоновану модель автоматизованого забезпечення екологічної рівноваги в сфері переробки відходів за оптимізаційною моделлю. Практична цінність роботи полягає в тому, що при застосуванні методу переробки відходів виконується не тільки відкидання параметрично неприйнятних характеристик відходів, а й запити користувачу на встановлення параметрів характеристик потенційно можливих заходів, що дозволяє в процесі роботи методу розширити картину взаємодії та взяти до розгляду апріорі неприйнятні параметри. При зворотному використанні методу, можливе прогнозування організаційно-виробничих заходів у сфері переробки відходів та формування шаблонів запитів на заходи, які формуються за відповідним обсягом прогнозованих характеристик відходів.

#### **10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота**

Рекомендую допустити кваліфікаційну роботу магістра до захисту.

Робота заслуговує на оцінку « задовільно ».

Науковий керівник \_\_\_\_\_



к.фіз.-мат.н., доц. Віталій Міхалевський