

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка системи для ідентифікації аномальних даних

Виконав: студент IV курсу, групи СП-41

спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Антонюк Д.Т.

(підпис)

(прізвище та ініціали)

Керівник

Пастух О.А.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Стоянов Ю.М.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль

2024

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

Петрик М.Р.

(підпис)

(прізвище та ініціали)

« » _____ 2024р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня	бакалавр
	(назва освітнього ступеня)
за спеціальністю	121 Інженерія програмного забезпечення
	(шифр і назва спеціальності)
студенту	Антонюк Дмитро Тарасович
	(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи для ідентифікації аномальних даних

Керівник роботи Пастух Олег Анатолійович, д.т.н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 2024 року № _____.

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Числовий масив даних – різні характеристики об'єктів нерухомості, наприклад, площа нерухомості, кількість кімнат, тощо; програмні засоби: PCA, KNN, LOF, OCSVM, IForest, Python and etc.

4. Зміст роботи (перелік питань, які потрібно розробити) Конструктивний аналіз програмного забезпечення для сфери ринку нерухомості, зокрема, для ріелторських компаній, що працюють з різними типами об'єктів нерухомості. Проектування та прикладне конструювання програмного забезпечення для аналізу ринку об'єктів нерухомості.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів) Проаналізовано актуальність теми роботи, мету та завдання роботи. Представлено результати кваліфікаційної роботи проектування програмного забезпечення для аналізу аномалій на ринку об'єктів нерухомості - детектування не типових об'єктів нерухомості, зокрема, виявлення об'єктів нерухомості з нестандартними характеристиками, наприклад, ціна, локація об'єкту нерухомості, тощо. Результати у вигляді програмного коду, даних, графіків, тощо. Зроблено метрологічний аналіз програмного забезпечення, який характеризує якість роботи програмної системи: коефіцієнт аномальності, міра рівня аномальності, які у своїй сукупності дають можливість виконати системні висновки щодо нештаних об'єктів нерухомості у базі даних. Організовано можливість зберігати прогр. компоненти розроблені програмні засоби для повторного їх використання. Проведено висновки по роботі.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Лазарюк В.В., к.т.н., доц., доц. каф. МТ		
Нормоконтроль	Стоянов Ю.М., к.т.н.		

7. Дата видачі завдання

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Вибір та затвердження теми		виконано
2.	Аналіз предметної області та технічного завдання		виконано
3.	Аналіз особливостей проектування та конструювання ПС		виконано
4.	Проектування програмної системи		виконано
5.	Практичне конструювання програмної системи		виконано
6.	Оформлення записки кваліфікаційної роботи		виконано
7.	Безпека життєдіяльності, основи охорони праці		виконано
8.	Підготовка презентації і доповіді		виконано
9.	Нормоконтроль		виконано
10.	Перевірка на плагіат		виконано
11.	Попередній захист кваліфікаційної роботи		виконано
12.	Захист кваліфікаційної роботи		виконано

Студент

Антонюк Д.Т.

(підпис)

(прізвище та ініціали)

Керівник роботи

Пастух О.А.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

У зв'язку з тим, що здійснюється перехід людства від інформаційних технологій до технологій даних, то у роботі розглянуто підхід до розробки програмного забезпечення на основі інтелектуалізованих алгоритмів, зокрема, алгоритмів машинного навчання: однокласового методу опорних векторів, ізольованого лісу, методу головних компонент, різного роду статистичних критеріїв, зокрема, статистичного критерію Грабба. Також у даній роботі виконано консолідацію цих алгоритмів машинного навчання, що наведені вище на основі їх голосування по принципу простого голосування. Тобто, в основі програмного забезпечення виконано ансамблювання вище наведених алгоритмів машинного навчання на основі простого голосування для вирішення прикладного завдання – детектування аномальних даних у наперед заданому наборі даних. Під аномальними даними у даній роботі маються на увазі дані, які представляють собою статистичні викиди у вибірці даних, саме тому використовуються статистичні тести – статистичний тест Грабба. Технологічні інструменти, які використані у даній роботі – це є програмні інструменти, наприклад, технології програмування пайтон та ряду бібліотек із дистрибутиву анаконди таких, як scikit-learn, pyod, pandas, numpy, тощо.

Теоретичною основою роботи є використання основної топологічної гіпотези, що статистичні викиди, тобто аномальні дані у наборі даних (вибірці даних) топологічно є ізольованими у конфігураційному (фазовому) просторі даних у якому координатні вісі інтерпритуються, як характеристики даних.

Предметна область для якої розглядається застосування вище наведеної програмної розробки – це детектування аномальних об'єктів нерухомості на ринку нерухомості.

ANNOTATION

In connection with the fact that humanity is transitioning from information technologies to data technologies, the paper considers an approach to software development based on intellectualized algorithms, in particular, machine learning algorithms: one-class method of support vectors, isolated forest, method of principal components, various statistical criteria, in particular, Grubb's statistical criterion. Also, in this work, the consolidation of these machine learning algorithms, which are given above, is performed on the basis of their voting according to the principle of simple voting. That is, the basis of the software is the ensemble of the above-mentioned machine learning algorithms based on simple voting to solve the applied task of detecting anomalous data in a predetermined data set. Anomalous data in this paper refers to data that represent statistical outliers in the data sample, which is why statistical tests are used - Grubb's statistical test. The technological tools used in this work are software tools, such as Python programming technologies and a number of libraries from the Anaconda distribution, such as scikit-learn, pyod, pandas, numpy, etc.

The theoretical basis of the work is the use of the main topological hypothesis that statistical outliers, i.e. anomalous data in a data set (data sample) are topologically isolated in the configurational (phase) data space in which coordinate axes are interpreted as data characteristics.

The subject area for which the application of the above software development is considered is the detection of anomalous real estate objects on the real estate market.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП.....	9
РОЗДІЛ 1. ДЕТЕКТУВАННЯ АНОМАЛЬНИХ ДАНИХ ДЛЯ ДОМЕНУ НЕРУХОМОСТІ.....	10
1.1. Домени у яких необхідно детектовувати не стандартні дані.....	10
1.2. Візуалізаційні та вербальні представлення виявлення аномальних даних.....	12
1.3. Зображення аномальних (не стандартних) даних.....	17
1.4. Мета, прикладні завдання кваліфікаційної роботи.....	19
РОЗДІЛ 2. ПРАКТИЧНО-ПРИКЛАДНЕ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДЕТЕКТУВАННЯ ДАНИХ.....	21
2.1. Реалізація на практиці архітектури.....	21
2.2. Імпортування бібліотек з дистрибутиву анаконди.....	24
2.3. Використання PCA для зображення багатовимірних даних.....	27
2.4. Проектування детектування на основі інструкцій статистичних критеріїв.....	28
2.5. Проектування детектування на основі інструкцій найближчих сусідів.....	30
2.6. Проектування детектування на основі інструкції LOF.....	32
2.7. Проектування детектування на основі інструкції PCA.....	33
2.8. Проектування детектування на основі інструкції IForest.....	36
2.9. Проектування детектування на основі інструкції ансамблювання.....	38
РОЗДІЛ 3. ПРАКТИЧНО-ПРИКЛАДНЕ КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ НЕ ТИПОВИХ ДАНИХ.....	39

3.1. Уведення бібліотек та фреймворків пайтона із дистрибутиву анаконди.....	39
3.2. Уведення даних нерухомості із зовнішнього середовища.....	40
3.3. Зображення багатовимірних даних на основі PCA.....	42
3.4. Практичне конструювання детектування на основі інструкцій статистичних критеріїв.....	43
3.5. Практичне конструювання детектування на основі інструкцій найближчих сусідів.....	44
3.6. Практичне конструювання детектування на основі інструкції LOF....	47
3.7. Практичне конструювання детектування на основі інструкції PCA....	49
3.8. Практичне конструювання детектування на основі інструкції OCSVM.....	51
3.9. Практичне конструювання детектування на основі інструкції IForest	52
3.10. Практичне конструювання детектування на основі інструкції ансамблювання.....	54
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	56
4.1. Аварії з викидом радіоактивних речовин.....	56
4.2. Техніка безпеки при проведенні електричних вимірювань та випробувань установки.....	59
ВИСНОВКИ.....	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66
ДОДАТКИ.....	68
Додаток А Технічне завдання.....	69
Додаток Б Тези доповіді на конференції.....	74

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

outlier – аномальний елемент множини даних

outlier score – функція оцінки аномальності елемента даних

inlier – нормальний елемент множини даних

PCA – метод головних (принципових) компонент

kNN – метод найближчих сусідів

LOF – фіча локального викиду

OCSVM (One-class SVM) – однокласовий метод опорних векторів

IForest – клас ізольованого лісу

ВСТУП

Поступове та все більше зміщення пріоритету функції уваги у життєдіяльності людства від інформаційних технологій до технологій даних спричиняє значні зміни у сфері інженерії програмного забезпечення. Зокрема, на передній план лавиноподібно починає виходити інтелектуалізоване програмне забезпечення, тобто таке, що у значній мірі використовує засоби штучного інтелекту.

Однією із важливих сфер застосування інтелектуалізованого програмного забезпечення є детектування аномальних даних у тих чи інших наборах даних чи базах даних. Сюди можна віднести, наприклад, технічну діагностику, медичну діагностику, виявлення кібератак, виявлення не стандартних гравців на фінансовому ринку чи інших видах ринків, тощо.

У цій роботі розглядається завдання виявлення аномальних даних на ринку нерухомості, тобто детектування не типових видів нерухомості: квартир, будинків, тощо.

Для цього у роботі спершу виконано проектування, а потім прикладне конструювання програмного забезпечення, що використовує інтелектуалізовані алгоритми, нахталт, найближчих сусідів, ізольований ліс, метод головних компонент, однокласовий метод опорних векторів, тощо, а також їх консолідацію у вигляді ансамблювання на основі голосування.

РОЗДІЛ 1. ДЕТЕКТУВАННЯ АНОМАЛЬНИХ ДАНИХ ДЛЯ ДОМЕНУ НЕРУХОМОСТІ

Аналіз даних у наперед заданому їх наборі наразі представляє собою трендовий напрям, оскільки вагова функція уваги поступово, але невпинно зміщається з інформаційних технологій (ІТ) в сторону технологій даних (DT).

Оскільки зараз ми живемо в еру потоків великих даних, то їх обробка, аналіз повинні виконуватися із врахування цих моментів. Потрібно використовувати різного роду високопродуктивні обчислювальні інфраструктури, наприклад, кластеризованих обчислювальних систем, які можуть знаходитися віддалено. Керувати їх обчислювальними гіперпараметрами, зокрема, кількістю фізичних ядер, кількістю логічних ядер – потоків, кількістю ресурсів локальної пам'яті для окремої робочої комп'ютерної станції у обчислювальному кластері.

У сфері технологій даних однією із ключових задач є задача щодо виявлення не стандартних даних у тих чи інших наборах даних, тобто виявлення статистичних викидів у вибірковій сукупності даних, якщо висловлюватися у термінах математичної статистики, або ж детектування аномальних даних у множині даних.

Якщо розглядати напрям детектування аномальних даних, то слід відзначити, що під детектуванням мається на увазі два напрями – це піднапрямок – виявлення нових даних і другий напрям – це виявлення не штатних даних.

У даній роботі розглядається другий із піднапрямів – це ідентифікація не штатних, не шаблонних даних у наборі даних. Зокрема, у цій бакалаврській роботі розглядається галузь ринку нерухомості. Тобто необхідно виявити не стандартні об'єкти нерухомості у множині даних квартир, будинків, тощо, що є елементами нерухомості ринку.

На сьогодні створена велика лінійка різного роду програмно-інформаційно-аналітичних систем для вирішення такого завдання, тобто встановлення не стандартних даних у їх множині. В основі всі ці системи використовують основну

топологічну гіпотезу математичної сфери топології, зокрема, яка говорить, що дані в межах класів повинні бути топологічно близькими, а не стандартні дані, наприклад, якісь шахрайські об'єкти нерухомості на ринку є з точки зору математичної статистики – статистичними викидами, а значить вони є ізольованими точками у топологічному просторі множини даних об'єктів нерухомості.

Аналіз наявних програмно-інформаційно-аналітичних систем для опрацювання інформації щодо нерухомих сутностей показує, що існує ціла низка програмно-інформаційно-аналітичних систем: Real Estate CRM, IST.Realtor, Plektan, Realtsoft, тощо.

В розрізі даного огляду доречно розглянути роботи [1-9], які в неявній формі стосуються ряду наукових елементів, що використовуватимуться у цій роботі.

1.1 Домени у яких необхідно детектувати не стандартні дані

Завдання для детектування аномальних (не стандартних) даних розглядаються в наступних сферах людської діяльності:

- медична діагностика;
- ідентифікація користувачів та ботів;
- ідентифікація підозрілих банківських операцій;
- виявлення атак;
- виявлення не стандартних гравців на фінансовому ринку;
- технічна діагностика;
- сейсмологія.

Можна навести багато інших сфер застосування, де необхідно виявляти не нормальні дані у їх наборах, особливо коли мова йде про потоки великих наборів даних. Тому потрібно швидкі і надійні алгоритми для ідентифікації не нормальних, не стандартних, аномальних даних. Існує багато алгоритмічних підходів для

виявлення таких даних і всі вони містять різні різновиди алгоритмів, кожен з яких має, як свої переваги так і недоліки, наприклад щодо точності – функції, яка оцінює міру аномальності елементів множини даних.

1.2 Візуалізаційні та вербальні представлення виявлення аномальних даних

Як було написано вище в основі різних алгоритмів та їх консоладації для ідентифікації не нормальних даних лежить гіпотеза топологічної ізолюваності не стандартних даних у їх фазовому просторі. Для наочності розуміння розгляньмо рисунок нище представлений.

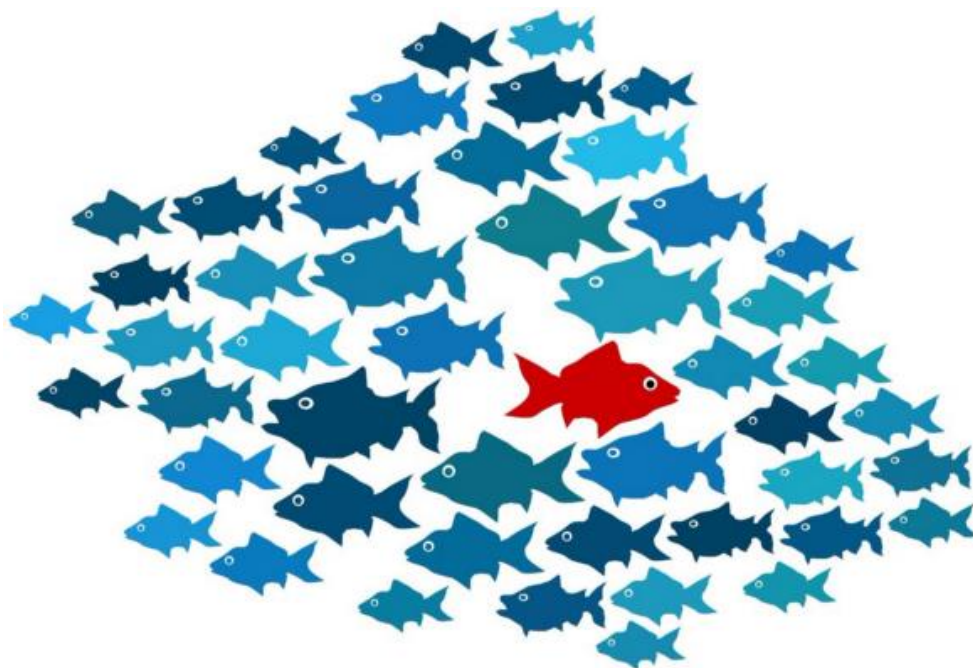


Рис.1.1 – Приклад анномальних даних

Тут під топологічної ізолюваністю у даних наведено атрибут кольору риб, як видно усі риби є різними відтінками синього, темно синього, голубого кольорів, а топологічн ізолювана точка у просторі даних представляється червоною рибою,

тобто фактично представляє собою не стандартні дані, тобто по координатній фічі елемента даних – риби є аномальною величиною із спектру значень кольорів, проте не лише так можна для наочності візуалізувати аномальність даних, накшталт, відокремлених точок у просторі даних. Можна розглянути і рисунок нище.

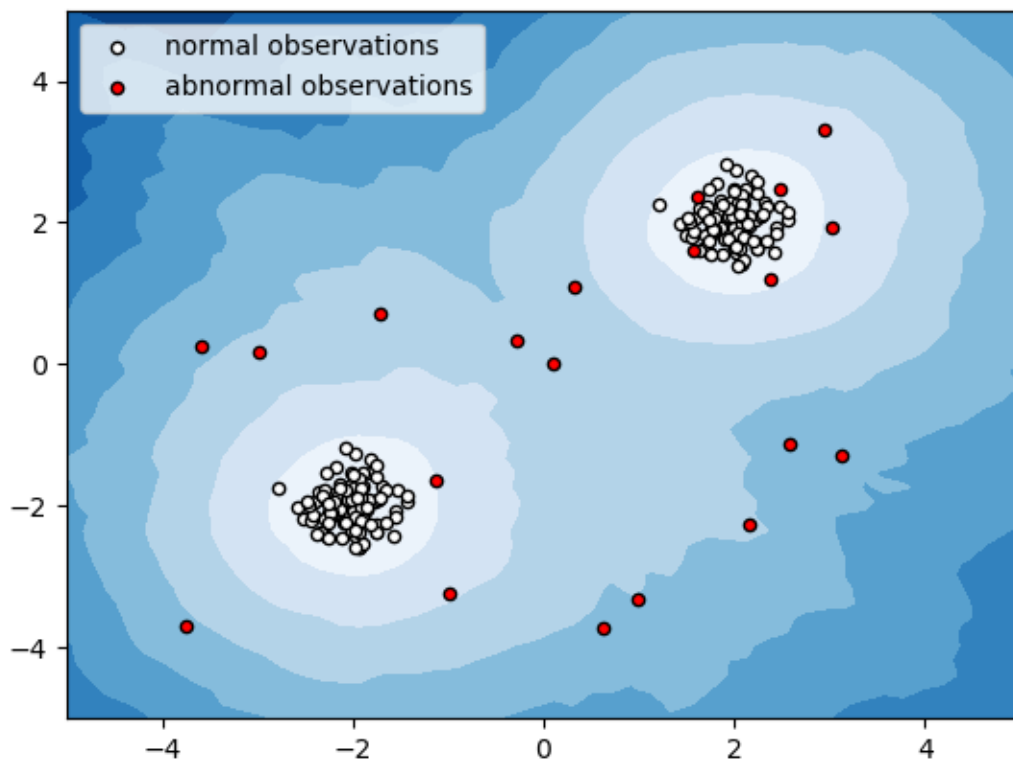


Рис.1.2 – Приклад анномальних даних у просторі даних

На даному рисунку видно, що нормальними точками даних є білі точки і вони топологічно знаходяться близько одна відносно одної, а не нормальні дані помічені у цьому просторі даних червоним кольором, вони і є аномальними даними.

Не стандартні дані зустрічаються у різного роду множинах даних, тобто множинах даних, які мають різну природу та інтерпритацію. Дані можуть мати різну кількість атрибутів, навіть сотні тисяч чи навіть мільйони. Всерівно, коли мова йде про не стандартні дані, то мається на увазі, що вони у фазовому просторі, конфігураційному просторі є у вигляді топологічно ізольованих точок, також правильно інтерпритувати їх, як ізольовані кінці векторів у векторному просторі, оскільки всі елементи множини даних завжди можна розглядати, як вектори, початки

яких співпадають і знаходяться у початку декартової системи координат не залежно від її розмірності, зауважимо, що атрибути даних чи компоненти таких векторів є окремими координатними вісями у цій декартовій системі координат. Кінці цих елементів даних, тобто їх векторних представлень знаходяться в різних точках даного простору. Тому відповідні кінці векторів, які відповідають елементам множини даних, які є не стандартними завжди будуть сильно віддаленими від загальної сукупності кінців векторів, що відповідають всім іншим елементам множини даних фіксованого набору.

Вище описані речі мають місце для довільної природи даних і не залежать від предметної області до якої вони відносяться. Так далі по тексту зображено рисунком аномалії амплітудних значень для часових послідовностей.

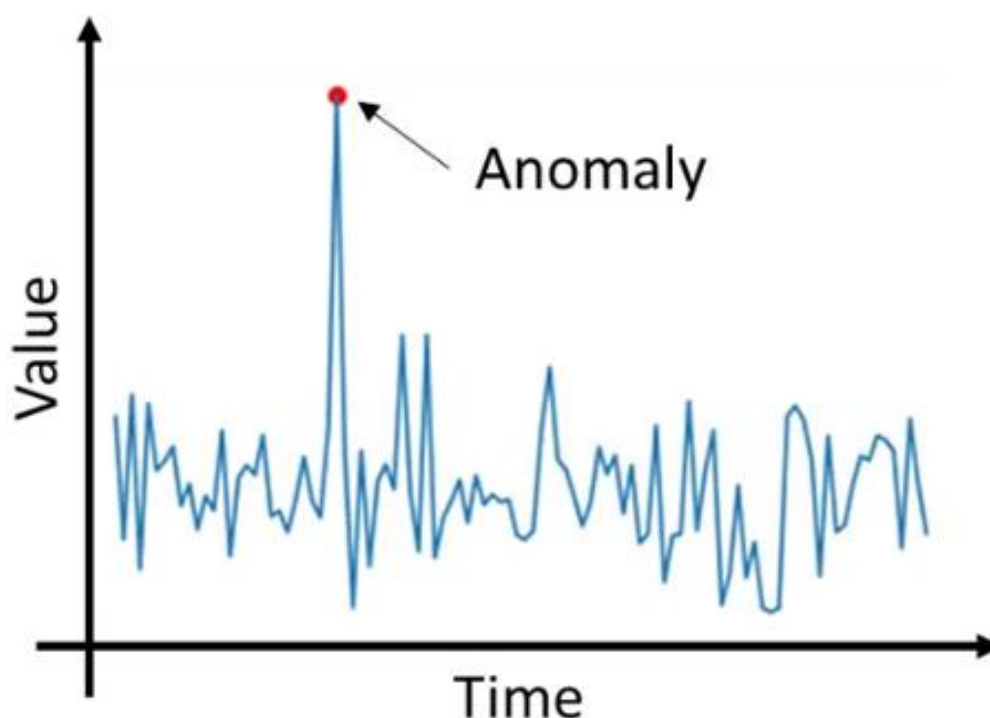


Рис.1.3 – Приклад анномальних даних в амплітуді часових рядів

Як видно із графічного зображення амплітуди часового процесу мають локалізовану варіаційну локацію на інтервалі вісі ординат, області значення даного процесу часу. Але разом з тим червоним кольором візуалізовано у фіксований момент часу амплітуду – миттєве значення процесу часу і це значення суттєво, тобто

значимо відрізняється від усіх інших миттєвих значень процесу часу, тобто дане значення є ізольованим від інтервалу варіювання усіх інших миттєвих значень процесу, що розгортається, як послідовність.

Аналогічні міркування можна застосувати і для зображення, що є нище наведеним і також представляє змінний по абсцисі процес.

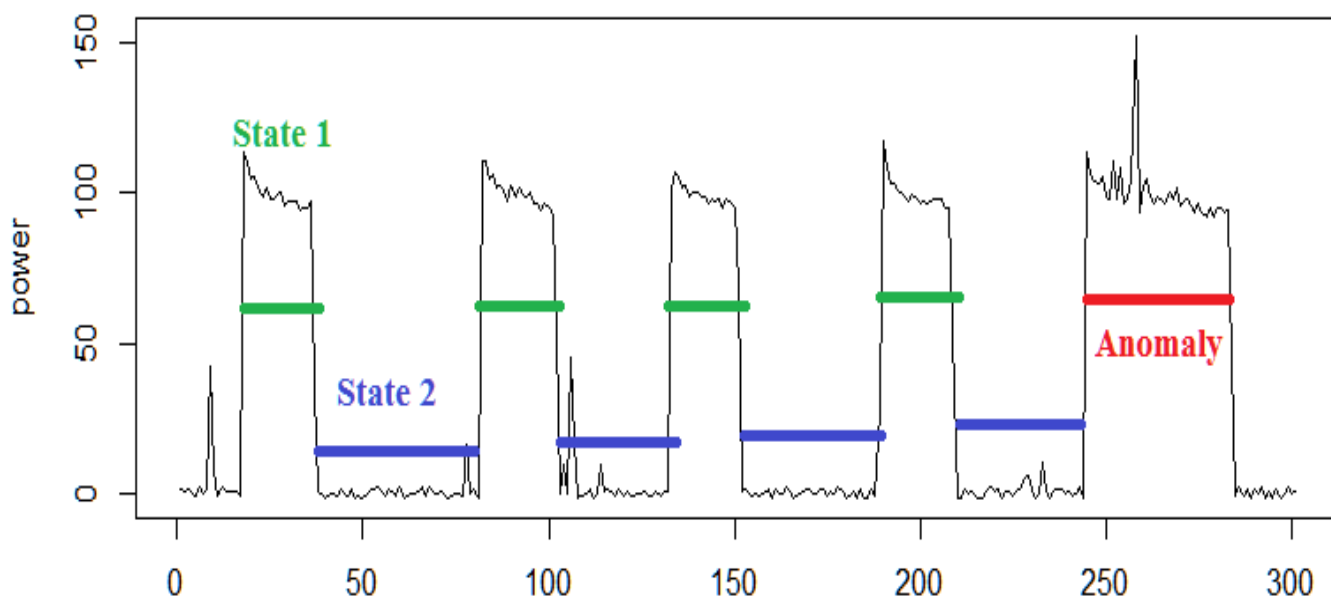


Рис.1.4 – Приклад анномальних даних у тривалостях часових рядів

Легко видно, що процес має типу періодичну форму, однак не все просто. Більшість тривалостей, що намальовані синім кольором у статистичному розумінні є на мові статистики не значимо відрізняємі, а значить топологічно локалізуються у певному ординатному інтервалі області значення послідовності розвиваючоїся у часі. Червоним кольором намальовано інтервальне значення, типу період, який фіксований в деякі моменти часового інтервалу і видно, що його числове значення, знову таки вживаючи статистичні поняття, є значимо відрізняючимся від усіх інших, а тому у топологічному фазовому просторі чи топологічному конфігураційному просторі це числове значення буде топологічно ізольованим, а значить задовольняє основному постулату, що стверджує – не стандартні дані, тобто анномальні є суттєво віддаленими від всіх інших значень у згустку чи то кластері значень усіх інтервальних періодів розглядуваної послідовності, яка розгортається в часі. Варто

тут сказати, що у просторах даних, де координатні вісі декартової системи координат інтерпритуються, як атрибути даних відстані можуть бути обчислені по різним формулам. По-перше вони всі утворюють метричні простора з різними формулами розрахунку відстані, наприклад, відстань у метричному просторі, що вираховується по формулі евкліда або відстань, що вираховується по формулі манхеттин, чи відстань, яка рахується, як відстань мінковського і таких прикладів можна навести багато, але ми тут у роботі їх більше не будемо приводити.

На наступному графічному зображенні наведено приклад того, що як написано раніше дані можуть мати різну природу, різну інтерпритацію, також мати різну структуру чи форму. Отже розглянемо нище форму даних у вигляді графової структури, яких тут є два види.

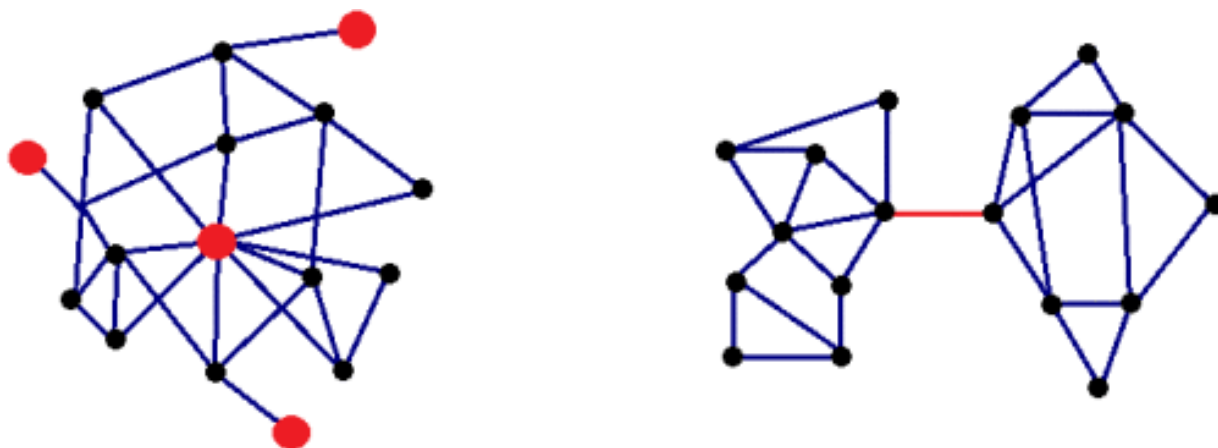


Рис.1.5 – Приклад анномальних даних у графових структурах даних

Аналогічно до попередніх прикладів, що були описані тут форма графових структур також може бути двох суттєво різних видів. У першому випадку під не стандартними чи анномальними елементами даних є вершини графової структури, тобто точки. Ті вершини, які намальовані синім кольором це є нормальні елементи множини даних і видно, що вони знаходяться у середині графу, а ізольовані чи анномальні елементи множини, тобто дані, що намальовані червоним кольором є крайніми точками, тобто вершинами чи їх ще називають нодами графової форми. Однак видно одну червону точку, один елемент, який розташований всередині

графової форми, це означає, що він є не нормальним в розумінні ступені цієї точки-вершини графу, оскільки в нього є набагато більше інцидентних ребер в порівнянні з іншими нодами-вершинами цього графу.

Також на цьому малюнку графічно справа наведений ще один граф-об'єкт, але вже в нього розглядається інший елемент, що намальований червоним кольором і представляє собою ребро. Це ребро є не нормальним у розумінні того, що воно зв'язує два графи в один великий, типу метаграф.

Розглянемо ще інші приклади, які зустрічаються дуже часто у прикладних застосуваннях, а саме, розглянемо рисунок нище, що зображає текстову випадкову упорядковану множину – послідовність.

AAABVSSAABVVSAABVVCABVSSAABABVSSAABABVVVVCABVVSS

Рис.1.6 – Приклад анномальних даних у текстових даних

Можуть бути не нормальні елементи даних і у символічних упорядкованих чи частково упорядкованих послідовностях, що представляють собою ґратки з точки зору алгебраїчних структур. Наприклад сюди можна віднести чи то частоту того чи іншого символу, чи то упорядкованість якихось символів, чи співпадання символів-літер, як наведено на вище малюнку, тобто типів прикладів для послідовностей не нормальних елементів даних – анномальних може бути велика кількість.

1.3 Зображення анномальних (не стандартних) даних

Але приводячи різні частинні приклади не нормальних представників – елементів даних того чи інакшого його набору узагальненої картині однаково не буде. Для формування узагальненого бачення потрібно перейти до абстракцій

вищого рівня щодо тлумачення аномальності у сукупності елементів даних. Потрібно оперувати поняттями, які вже згадувалися раніше, наприклад, фазовий простір набору даних чи то конфігураційний простір набору даних. Такі абстрактні поняття дають змогу сформувати узагальнене відчуття, яке покриє всі раніше приведені приклади з різних доменів народного господарства людей.

Розглянемо графічне зображення нижче і на основі нього приведемо тлумачення узагальнених понять для дефініції аномальності у даних.

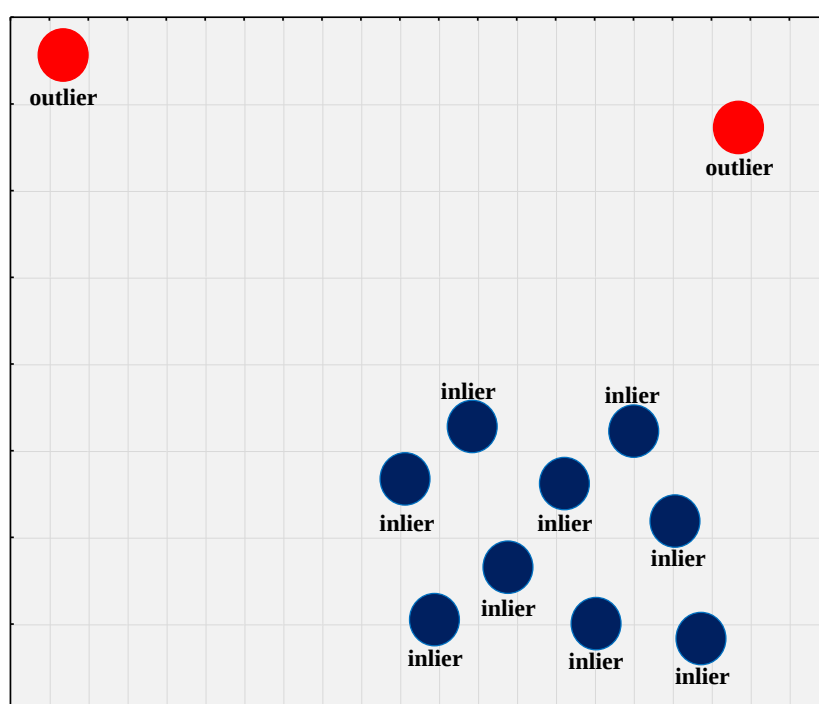


Рис.1.7 – Узагальн. зображ. основ топологічн. ізолюваності точок простору даних

Для прикладу тут наведено двовимірний простір даних, тобто коли дані із набору мають дві фічі чи атрибути. Загалом по індукції можна собі представити і багатовимірний простір з розмірністю мільйони та мільярди значної величини.

З малюнку видно, що кластер даних, які намальовані синім кольором утворюють топологічний згусток, тобто як і звучить в основній топологічній гіпотезі нормальні дані повинні кучкуватися, тобто розташовуватися близько один відносно одного, це і спостерігається на даному малюнку. Червоним кольором намальовані дві точки, які є ізолюваними, тобто на статистичній мові є викидами, а на мові даних

є аномальними даними відносно усіх інших елементів множини даних. Видно, що вони знаходяться у розумінні відстані достатньо віддалено від всіх решта даних, що знову вкладається у дефініцію основної топологічної гіпотези про аномальність в наборі даних.

У даній роботі є наявний набір даних з інтернет платформи кагл, де відбуваються змагання для інженерів у галузі машинного дослідження їх навчання. Набір – множина даних стосується різного роду нерухомості, типу квартири, будинки, тощо у пригороді міста Бостон. Завдання, яке розглядається – це ідентифікувати ненормальну нерухомість в множині даних, наприклад, нерухомість у якої необ'єктивна ціна, тобто занадтно завищена або занадто занижена, або має довільні інші фічі, що суттєво її відрізняють від всіх інших видів нерухомості і цим самим вона є елементом викиду у вибірці даних – аномалією, яку потрібно детектувати з цього набору-множини даних.

Тому потрібно розробити софт для виявлення ненормальних об'єктів нерухомості на ринку, щоб зменшити ризики для клієнтів, які планують її придбати, тобто, щоб уникнути потенційних шахрайських схем.

1.4 Мета, прикладні завдання кваліфікаційної роботи

Перейдемо до розгляду формулювання мети у цій роботі бакалавра та завдань, які будуть вирішуватися у межах цього дослідження.

Мета.

Концептуально спроектувати та практично сконструювати (практично реалізувати, практично розробити) програмні засоби для детектування (ідентифікації) аномальних (не стандартних) даних у наборі об'єктів нерухомості.

Завдання.

1. Концептуально спроектувати програмні засоби для детектування (ідентифікації) аномальних (не стандартних) даних у наборі об'єктів нерухомості.
2. Практично сконструювати (практично реалізувати, практично розробити) програмні засоби для детектування (ідентифікації) аномальних (не стандартних) даних у наборі об'єктів нерухомості.

РОЗДІЛ 2. ПРАКТИЧНО-ПРИКЛАДНЕ ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДЕТЕКТУВАННЯ ДАНИХ

Одним із центральних підходів при розробці програмного забезпечення для різного плану технічних систем є етап проектування. Цей етап завжди передує етапу практичної розробки. І чим складніша технічна система планується розробляється тим важливішим стає етап проектування, оскільки він дає змогу зпланувати багато різних нюансів при подальшій практичній розробці.

Тому важливість цього розділу є дуже великою, оскільки він стосується етапу проектування програмного забезпечення для техніко-інформаційної системи, що облікує різного типу об'єкти нерухомості, реалізовує по них пошук, може виявляти різного роду ненормальні об'єкти нерухомості, тобто аномальні об'єкти нерухомості. І від якості проектування у цьому розділі, деталювання різного роду елементів програмного забезпечення для ідентифікації специфічно аномальних елементів даних буде залежати значення функції аномальності, яка визначає степінь – імовірність того, що елемент потрапляє у зону підозрілих елементів, тобто таких, який не подібний на інші елементи даних.

2.1 Реалізація на практиці архітектури

Практично-прикладне проектування програмного забезпечення для розроблюваної програмно-інформаційно-аналітичної технічної системи щодо виявлення нештатних атрибутів, тобто їх значень в окремих елементів даних виділяє

їх серед всіх інших елементів даних сукупності і цим самим получається, що вони є аномальними. Для ефективної розробки, тобто розробки, яка би давала високу точність на основі функції аномальності, високої надійності, що забезпечувало би високу прицизійність у роботі розглянемог узагальнену архітектуру, яка візуалізована нище і детально опишемо її функціонування на загальному рівні та на рівні блоків архітектури.

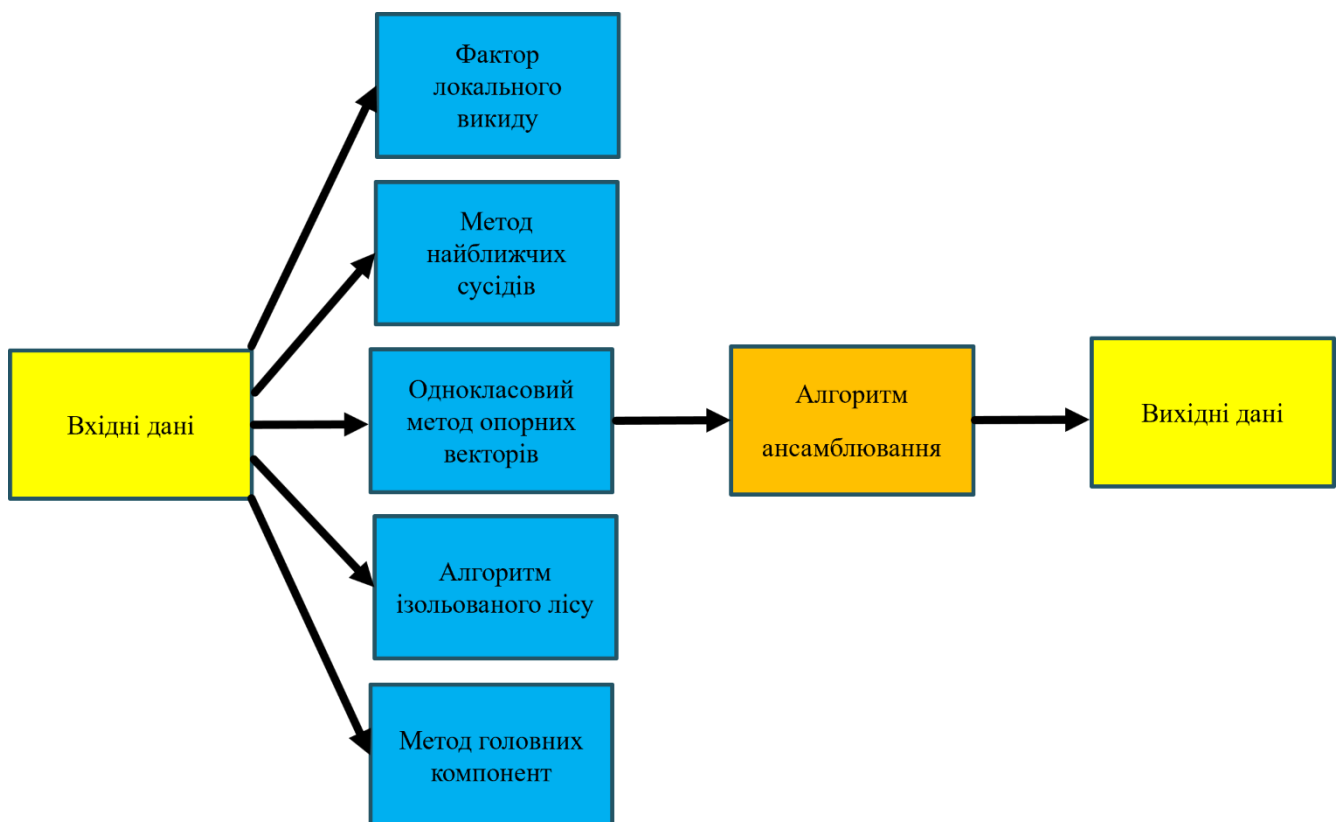


Рис.2.1 – Архітектура програмн. забезпечен. на основі ансамблювання алгоритмів

Узагальнена архітектура програмного забезпечення системи, що проектується містить ряд блоків, що віддзеркалюють роботу базових алгоритмів даної системи, типу ізольований ліс, метод головних компонент, тобто це різні алгоритми із галузі навчання машин. Сюди ж відносяться різні статистичні тести – статистичний тест Грабба. Також тут використовуються алгоритми однокласовий метод опорних векторів, можна використовувати інші довільні алгоритми, так звані метричні

алгоритми, що базуються на метричних векторних просторах, та й загалом на топологічних векторних просторах, а ще точніше на топологічно-метричних векторних просторах.

Відбувається розпаралелення вхідних даних по усіх цих алгоритмах, що відзначені у попередньому абзаці. Кожен із цих алгоритмів, методів повертає дані, які оцінив, як аномальні, а також для них повертає функцію, що оцінює міру аномальності того, що цей елемент відноситься до нестандартних даних.

Після цього дані від усіх алгоритмів та методів, що є у даній архітектурі потрапляють на алгоритм, який реалізує ансамблювання по методу простого голосування вихідних результатів всіх алгоритмів та методів і реалізує консолідований фідбек цілої системи.

Загалом, якщо розглядати такого роду архітектуру, то при проєтуванні по цьому принципу варто використовувати якомога найрізноманітніші базові алгоритми, щоб вони завдяки своїй різноманітній природі покрили якнайширший спектр різнобарв'я атрибутів даних.

Ансамблювання при цьому проєтуванні архітектури може бути організовано у найрізноманітніші способи. Наприклад, можна у цій проєктованій архітектурі розглянути зважене голосування базових алгоритмів і за вагу голосу для кожного алгоритму обрати його точність прийняття рішень при його тестуванні на тестовій множині даних по принципу перехресної перевірки. Тобто, чим вища точність була у базового алгоритму тим більша його вага при голосуванні у сукупності решта алгоритмів і яку реалізує мета алгоритм шляхом ансамблювання по принципу зваженого голосування. Таким форматом реалізовується інший тип ансамблювання алгоритмів, що у проєктованій архітектурі знаходяться у базисі архітектури.

У цій роботі бакалавра розглядається випадок ансамблювання на іншій основі, тобто ансамблювання, яке використовує інше правило консолідації базових рішень. А саме, ансамблювання заплановано реалізувати на основі правила простого голосування основних – базових інструкцій, тобто алгоритмів. Тому цей вид

консолідованого рішення всіх алгоритмів не залежить від точності того чи іншого алгоритму, а все відбувається простим чином – приймається саме те рішення при голосуванні, яке прийняло більшість базових алгоритмів, що входять у процедуру голосування. Проста більшість визначає конкретний вид прийнятого рішення.

2.2 Імпортування бібліотек з дистрибутиву анаконди

Наступним етапом після проектування узагальненої архітектури програмного рішення завдання щодо ідентифікації ненормальних – аномальних даних серед різної множини даних об'єктів нерухомості важливе місце має використання тих чи інших програмних засобів. Важливість етапу проектування використання тих чи інших програмних засобів визначає багато функціональних характеристик програмного рішення. Наприклад, його масштабованість на комп'ютерні кластери, що реалізовані на хмарах, зокрема, масштабованість щодо кількості процесорний фізичних ядер, кількості використання логічних ядер, тобто потоків обчислення, кількості локальних ресурсів пам'яті у привязці до конкретного фізичного ядра. А це все визначає швидкодію роботи програмного рішення, масштабованість його на хмарні обчислювальні сервіси, тощо.

Також вибір програмного забезпечення суттєво впливає на його подальшу підтримку. Тобто якщо інструменти вибрані правильно – загально відомі і у вигляді обгортки, що забезпечують полегшення процесу проектування процесу безпосереднього програмування. На основі принципу бритви Оккама чим простіше реалізовано програмне забезпечення – доступними інструментами загально відомими, тим легше його підтримувати у подальшій роботі і цим самим здешевлюється його розробка та процес підтримки, оскільки легше забезпечити

людський ресурс для такого виду роботи. Тому тут обрано як основну бібліотеку – це пайоді, оскільки вона є обгорткою над самою доступною і поширенішою бібліотекою скітлерн. Звісно і використовується скітлерн безпосередньо на пряму, а також окрім них використовуються, як завжди, базові бібліотеки дистрибутиву анаконди – це бібліотеки для маніпулювання даними нумпай та пандас. Нище можна розглянути імпортування процес усіх перерахованих вище представників програмних інструментів у скрипт розробки.

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from pyod.models.knn import KNN
from pyod.models.lof import LOF
from pyod.models.pca import PCA as pyod_PCA
from pyod.models.ocsvm import OCSVM
from pyod.models.iforest import IForest
```

Основними алгоритмами, які в узагальненій архітектурі використовуються це є представник із бібліотеки пайоді, зокрема якщо говорити на мові бекенду, то клас із скітлерну – це метод головних компонент. Цей клас планується до використання у даній системі, оскільки він при проектування даних із багатовимірному простору їх атрибутів проектує дані у меншої розмірності простір, але при цьому згортає найбільше інформації.

Також використовується бібліотека пайтона матплотліб для візуалізації даних у двовимірному декартовому просторі. До основних алгоритмів ідентифікації аномальних даних у архітектуру включений метод найближчих сусідів, оскільки він відноситься до метричних методів і дозволяє краще у топологічному просторі даних відділяти ізольовані дані від усієї сукупності даних, що не представляють собою аномалій. Також використовується алгоритм ізольованого лісу, оскільки він володіє усіма сильними стоорнами випадкового лісу, нахштатт, випадковий вибір атрибутів, випадковий вибір елементів даних із вибірки і він, що є важливо дає можливість ізольованість виловлювати по довільному із атрибутів даних. Тобто на відміну від,

наприклад статистичних методів, які працюють на рівні окремих атрибутів, то ізольований ліс працює на довільному атрибуті і може аномалію виявити по довільній фічі даних. Але перейдемо до першого проектного процесу – процесу зчитування даних із зовнішнього середовища у скрипт проекту. Для цього у проекті, як очевидно, використовуватиметься маніпуляційна даними бібліотека – пандас.

```
pandas.read_csv(
    filepath_or_buffer, *, sep=_NoDefault.no_default,
    delimiter=None, header='infer',
    names=_NoDefault.no_default,
    index_col=None, usecols=None,
    dtype=None, engine=None,
    converters=None, true_values=None,
    false_values=None,
    skipinitialspace=False,
    skiprows=None, skipfooter=0,
    nrows=None, na_values=None,
    keep_default_na=True,
    na_filter=True,
    verbose=_NoDefault.no_default,
    skip_blank_lines=True, parse_dates=None,
    infer_datetime_format=_NoDefault.no_default,
    keep_date_col=_NoDefault.no_default,
    date_parser=_NoDefault.no_default,
    date_format=None, dayfirst=False,
    cache_dates=True, iterator=False,
    chunksize=None, compression='infer',
    thousands=None, decimal='.',
    lineterminator=None, quotechar='\"',
    quoting=0, doublequote=True,
    escapechar=None, comment=None,
    encoding=None, encoding_errors='strict',
    dialect=None, on_bad_lines='error',
    delim_whitespace=_NoDefault.no_default,
    low_memory=True, memory_map=False,
    float_precision=None, storage_options=None,
    dtype_backend=_NoDefault.no_default
)
```

Отже у цьому підрозділі даного розділу розглянуто етап проєтування узагальненої архітектури програмної реалізації аномалій житлових об'єктів, як даних із множини даних. Також розглянуто процес проєтування зчитування даних із зовні у скрипт проекту розробки. Перейдімо до наступного поетапного проєтування програмних компонент для подальшої їх практичної реалізації у коді скрипта для реалізації основних алгоритмів архітектури для детектування ненормальних – аномальних даних серед їх множини.

2.3 Використання PCA для зображення багатовимірних даних

Першим окремим кроком у роботі розглядається завдання візуалізації даних з метою розгляду їх топологічної структури і потенційної можливості побачити ізольовані точки – дані із набору. Для цього планується практична реалізація у коді на основі проєтування з використанням класу бібліотеки скітлерн, а зокрема методу головних компонент, що приведений нише по тексту із своїми гіперпараметрами для керування ним у програмній реалізації при безпосередньому кодуванні.

```
class sklearn.decomposition.PCA(
n_components=None,
*,
copy=True,
whiten=False,
svd_solver='auto',
tol=0.0,
iterated_power='auto',
n_oversamples=10,
power_iteration_normalizer='auto',
random_state=None
)
```

Головним гіперпараметром управління даним класом – це ж кількість компонент, оскільки саме вони визначають скількох розмірним простором даних будуть дані після його застосування. Оскільки розглядається випадок для візуалізації, то вибрано дві головні компоненти у цьому класі для його реалізації.

2.4 Проектування детектування на основі інструкцій статистичних критеріїв

Статистичні тести використовуються, як правило, у тих випадках, коли потрібно детектувати аномальні дані по окремим характеристикам. Статистичні тести детектують значення характеристик, які є екстримальними. При проектуванні програмного забезпечення для детектування аномальних даних використовують різні статистичні тести: статистичний тест Грабба, статистичний тест Шовена, статистичний тест Пірса, статистичний тест Діксона, а також інші статистичні тести. Проте найчастіше використовують статистичний тест Грабба. Статистика для даного тесту обчислюється за формулою.

$$\tau = \frac{|f_{extreme}(\vec{x}) - \hat{m}_{f(\vec{x})}|}{\hat{\sigma}_{f(\vec{x})}} = \frac{\max |f(\vec{x}) - \hat{m}_{f(\vec{x})}|}{\hat{\sigma}_{f(\vec{x})}}$$

Обчислювальні формули для нульової та першої статистичних гіпотез мають вид такий, що записаний далі. Загалом можна використовувати багато інших, різноманітних статистичних тестів, наприклад, статистичний тест Шовена чи статистичний тест Дейнса, та багато інших. Головне у них є керування ними на основі гіперпараметрами, зокрема, рівнем значимості, оскільки саме цей

гіперпараметр своїм числовим значенням задає похибку прийнятого рішення, коли обрана гіпотеза нульова, а значить накладаються сильніші умови на ідентифікацію аномальної нерухомості у наборі даних. Далі наведено формульні вирази для статистичних гіпотез статистичного тесту Граба.

$$H = \begin{cases} H_0, & \tau = \frac{|f_{extreme}(\vec{x}) - \hat{m}_{f(\vec{x})}|}{\hat{\sigma}_{f(\vec{x})}} = \frac{\max |f(\vec{x}) - \hat{m}_{f(\vec{x})}|}{\hat{\sigma}_{f(\vec{x})}} \leq \tau_{1-p}, \\ H_1, & \tau = \frac{|f_{extreme}(\vec{x}) - \hat{m}_{f(\vec{x})}|}{\hat{\sigma}_{f(\vec{x})}} = \frac{\max |f(\vec{x}) - \hat{m}_{f(\vec{x})}|}{\hat{\sigma}_{f(\vec{x})}} > \tau_{1-p}, \end{cases}$$

Для практичної кодової реалізації достатньо скористатися ресурсами пандас, оскільки як і нумпай так і пандас в основі своїй завжди працюють в паралельному режимі із змінними і це дуже добре, оскільки фактично вже забезпечено розпаралелення автоматично. Далі по тексту наведено кодову реалізацію статистичного тесту Граба у пандасі.

```
#Inliers
inliers_1 = df[((df-df.mean())/df.std()).abs()<3].dropna()
#Outliers
outliers_1 = df[((df-df.mean())/df.std()).abs()>3].dropna(how='all')
```

Як видно з кодового представлення тут обрано критерій три сигма, тобто обрано інтервал для значень, що задовольняють нульовій гіпотезі.

Таким видом коду реалізовано виділення значень відповідно, що задовольняють нульовій статистичній гіпотезі та виділено значення, які задовольняють статистичній гіпотезі ашодин. Рівень значимості при цьому, тобто альфа рівень дорівнює, як і у більшості випадків нуль цілих п'яти сотих – це значення імовірності для спрацювання нульової статистичної гіпотези у тесті Грабба. З кодового рішення видно, що відразу ж витираються ті рядки, які не задовольняють гіпотезі нульового рівня і при цьому спрацьовує перша гіпотеза. Безпосереднє

кодування Грабба тесту методами пандас є дуже ефективним, оскільки існують вже готові програмні рішення, на кшталт середньоквадратичного відхилення, обчислення середньоарифметичного значення., тощо.

2.5 Проектування детектування на основі інструкції найближчих сусідів

Наступним основним алгоритмом спроектованої архітектури є алгоритм найближчих сусідів. Цей алгоритм відноситься до класу метричних алгоритмів, тобто це означає, що він може керувати різними відстаннями у метричному просторі даних об'єктів нерухомості і відповідно розглядати у цьому метричному просторі ізольовані – аномальні дані нерухомості. Можна керувати гіперпараметру відстані, наприклад, відстань евкліда, відстань манхеттен, чи мінковського відстань, тощо. Саме алгоритм найближчих сусідів, як метричний алгоритм він найбільш інтерпретаційно близький до суті розв'язуваної задачі у цій бакалаврській роботі, тому він і був обраний у склад алгоритмів базових до спроектованої узагальненої архітектури програмної реалізації – забезпечення.

Метод найближчих сусідів має цілий ряд гіперпараметрів для свого управління, значення яких можна змінювати і цим самим керувати точністю рішення алгоритмом найближчих сусідів. Одним із головних гіперпараметрів є вид ядерної функції. Існує багато різновидів ядерної функції, наприклад, клас експоненційних ядерних функцій – гаусових, у вигляді розподілу імовірностей експоненційний закон та багато інших, що не належать до класу експоненційних ядер. Гіперпараметр ядерних функцій керує ваговою значимістю сусідів в залежності від відстані віддаленості сусідда до точки в якій розраховується вплив алгоритму. Як правило для завдань, на кшталт, ідентифікації ненормальних елементів даних – аномальних

найкращий результат, відомо із евристики, показує гаусовий вид ядерної функції. Нище на малюнку графічно наведено, як даний алгоритм відділяє внутрішні точки простору – нормальні елементи даних від ненормальних – аномальних елементів даних. Наведено два різних малюнки. Перед тим, як перейти до обґрунтування використовуваного програмного засобу для практичної кодової реалізації даного методу розглянемо ці два рисунки.

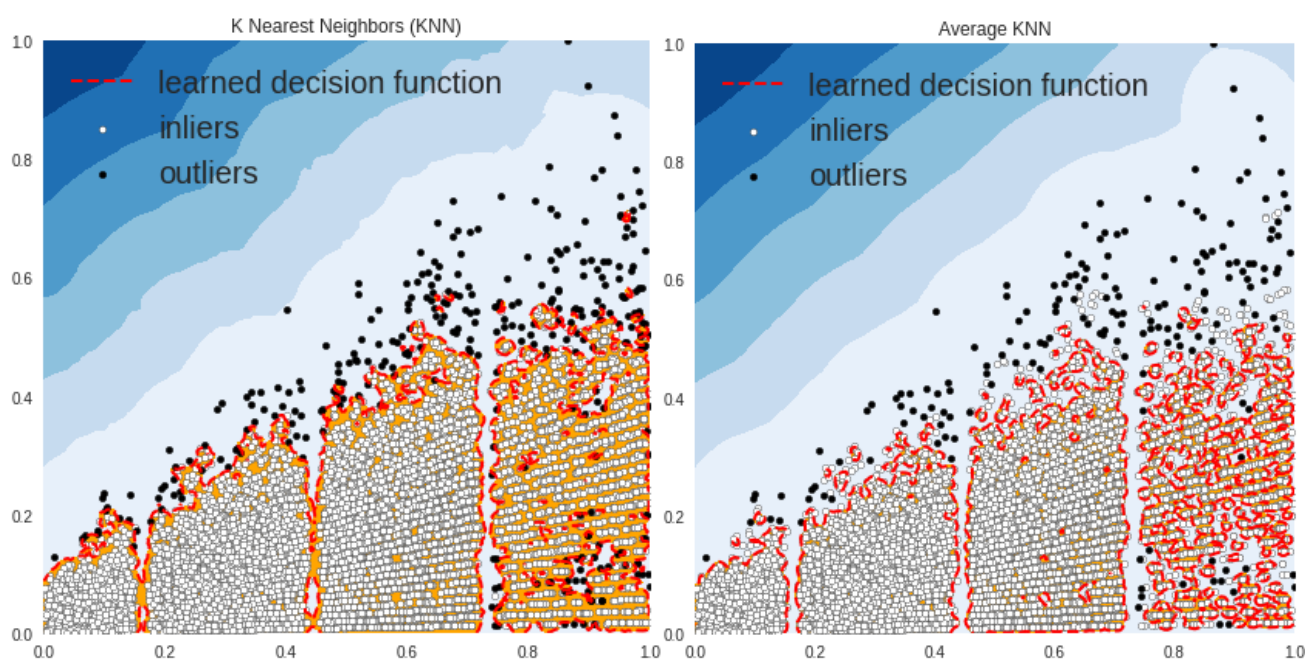


Рис.2.2 – Детектування аномальних даних на основі метричних методів

Для кодування цього алгоритму найближчих сусідів було вибрано використати обгортку у вигляді бібліотеки пайоді над бекендом у вигляді бібліотеки скітлерн, оскільки ця скітлерн програмний модуль є, по-перше, найбільш відомим, по-друге, він є розумно спроектованим, по-третє, розроблена на високому рівні якості та доступності технічна документація для фіксованого алгоритму. Ці всі перераховані переваги лягли в основу прийняття рішення щодо використання саме цього інструменту при проектуванні програмного рішення, як елемента у узагальненій архітектурі технічної розробки, що розглядається в даному розділі. Перед тим, як безпосередньо навести приклад програмного класу, який представляє

цей метод найближчих сусідів, лише черговий раз відзначимо, що він володіє іншими гіперпараметрами, які також впливають суттєво на якість його роботи при вирішенні різного типу прикладних задач.

```
class pyod.models.knn.KNN(
    contamination=0.1,
    n_neighbors=5,
    method='largest',
    radius=1.0,
    algorithm='auto',
    leaf_size=30,
    metric='minkowski',
    p=2,
    metric_params=None,
    n_jobs=1,
    **kwargs
)
```

На цьому з одним із основних алгоритмів у проектованій архітектурі – найближчих сусідів завершимо і перейдемо до розгляду наступного алгоритму, який входить до пулу основних – базових алгоритмів даної розробки.

2.6 Проектування детектування на основі інструкції LOF

За технічно-програмну основу реалізації у програмному кодуванні використовується та ж бібліотека, а саме пайоді, як обгортка, а також у її бекенді використовується скітлерн, оскільки вона у бекенді надає усі переваги даному алгоритму, як і всім іншим алгоритмам визначення аномалій, тобто узгоженість адаптивного програмного забезпечення, хороший рівень представлення документації, а значить полекшується, як безпосередньо розробка так і підтримка

програмного забезпечення, та й загалом його тестування. У символічно-програмному виді даний клас має вигляд, який наведений далі по тексту.

```
class pyod.models.lof.LOF(
n_neighbors=20,
algorithm='auto',
leaf_size=30,
metric='minkowski',
p=2,
metric_params=None,
contamination=0.1,
n_jobs=1,
novelty=True
)
```

Завершуючи проектний опис цього програмного класу, який може реалізувати дію одного із основних алгоритмів над яким виконуватиметься ансамблювання, перейдемо до наступного базового алгоритму для повноти процесу проектування.

2.7 Проектування детектування на основі інструкції PCA

Черговим базовим алгоритмом у проектуванні програмній частині технічної розробки є головних компонент метод. Даний метод має широке використання у вирішенні найрізноманітніших задач – зменшення розмірності простору даних, для їх візуалізації, виділення – екстракція мета інформативних фіч для даних, щоб система їх розуміла на вищому абстрактному рівні, фільтрації інформаційного шуму., тощо.

Перед тим, як перейти до візуалізаційних зображень, що полегшують його сприйняття та кодового його представлення звернімо увагу на те що даний головних

компонент метод використовується, як один із основних методів у проєктованій архітектурі і використовується його кон'юнкція властивостей, зокрема, фільтрація шумів, мета інтерпретація атрибутів даних і збільшення ізолюваності даних – об'єктів нерухомості, що суттєво відрізняються від всіх решта об'єктів нерухомості, що наявні у множині даних. Для візуальності нижче є ілюстрація.

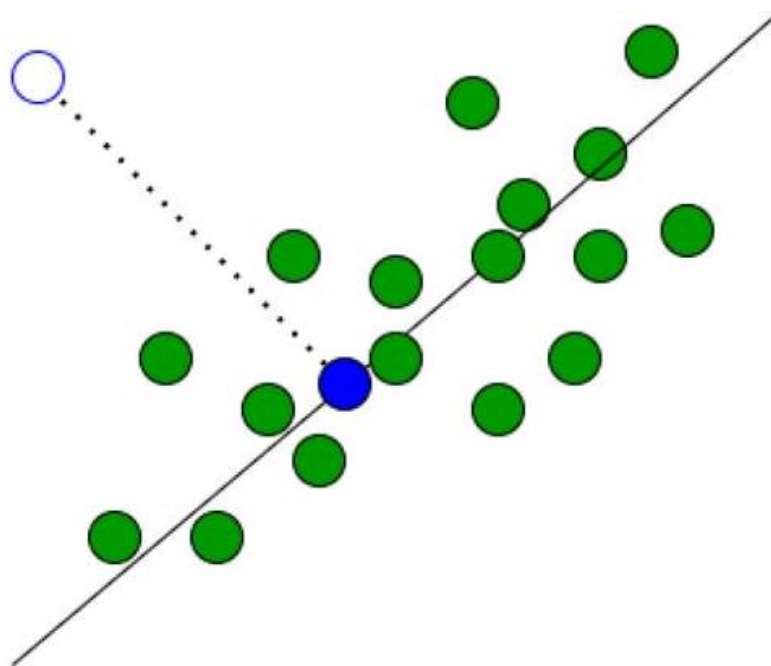


Рис.2.5 – Детектування аномальних даних на основі методу головних компонент

Точки зеленого кольору представляють нормальні об'єкти на ринку нерухомості, тобто не виходять по своїм сукупностям атрибутів за межі довірчих інтервалів вибірок з наперед заданою кількістю міри квантиля на рівні нуль цілих дев'яносто п'яти сотих.

Основним гіперпараметром для керування головних компонент методом є кількість головних компонент. У роботі при проєктуванні планується використати дві головні компоненти, по-перше їх можна візуалізувати – дані, нерухомість по друге завдяки подавленню інформаційного шуму в атрибутах даних краще ізолюються ті квартири, будинки, тощо, які суттєво по сукупності своїх атрибутів

відрізняються від усіх решта об'єктів нерухомості наявних в наборі даних. Для кодового рішення використовується бібліотека пайоді, як обгортка і скітлерн, як бекенд у даній технічній розробці. Розглянемо цей програмний клас.

```
class pyod.models.pca.pyod_PCA(
    n_components=None,
    n_selected_components=None,
    contamination=0.1,
    copy=True,
    whiten=False,
    svd_solver='auto',
    tol=0.0,
    iterated_power='auto',
    random_state=None,
    weighted=True,
    standardization=True
)
```

Окрім цього основного алгоритму проектованої програмної розробки використовується ще й інший алгоритм – мова про нього піде далі по тексту у наступному підрозділі.

2.8 Проектування детектування на основі інструкції OCSVM

Даний однокласовий опорних векторів метод є одним із також потужних алгоритмів, оскільки завдяки добре розробленій теорії він має велику кількість гіперпараметрів для свого керування. І одним із основних гіперпараметрів керування над ним є різновид, причому широкий різновид ядерних функцій, як лінійної так і різних типів нелінійності причому можна ще й керувати рівнем нелінійності ядерних нелінійних функцій і це суттєво вплине на точність роботи даного алгоритму по

детектуванню анномальних даних із сукупності даних. Для його практичної кодової реалізації використовується бібліотека пайоді із бекендом бібліотекою скікітlern, яка використовується у всіх інших попередніх основних алгоритмах проектованої розробки. Далі наведено програмний клас, який представляє цей однокласовий опорних векторів метод.

```
class pyod.models.ocsvm.OCSVM(
    kernel='rbf',
    degree=3,
    gamma='auto',
    coef0=0.0,
    tol=0.001,
    nu=0.5,
    shrinking=True,
    cache_size=200,
    verbose=False,
    max_iter=-1,
    contamination=0.1
)
```

Поряд із використанням даного алгоритму сильним є використання різновиду алгоритмів із логічної парадигми, а саме із випадкових лісів подібних це ізольований ліс. Він описаний у проектувальному процесі нище у наступному підрозділі.

2.8 Проектування детектування на основі інструкції IForest

Ізольований ліс включений у проект узагальненої архітектури по причині того, що він дає цілий ряд переваг у порівнянні з іншими алгоритмами. Наприклад, завдяки тому, що він випадковим чином вибирає атрибути даних він імовірно завдяки генератору псевдовипадкових чисел вибере необхідну фічу даних. Також

завдяки генератору псевдовипадкових чисел він випадково вибере аномальні дані – об'єкт нерухомості із усієї випадкової вибірки – набору даних. Всі аномальні об'єкти нерухомості будуть знаходитися у верхніх ярусах дерев цього ізольованого лісу. На малюнку наведено дерева цього алгоритму. Синім кольором позначення точки, які відповідають нормальним об'єктам нерухомості у множині елементів даних, а червоним кольором позначення викиди, тобто ненормальні – аномальні об'єкти нерухомості у наборі заданих даних.

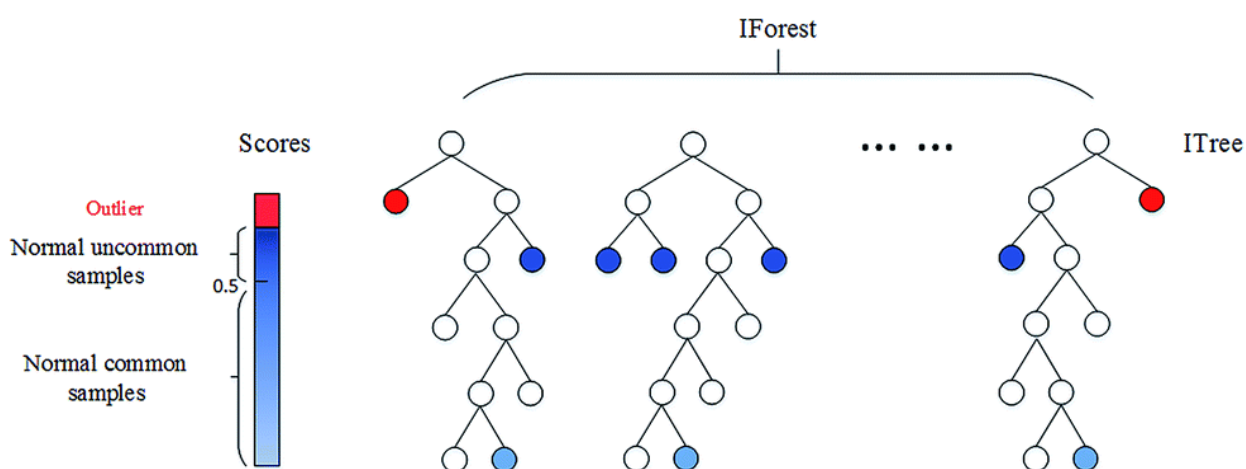


Рис.2.8 – Детектування аномальних даних на основі ізольованого лісу

Кодоке представлення ізольованого лісу наведено у вигляді програм. класу.

```
class pyod.models.dif.IForest(
    batch_size=1000,
    representation_dim=20,
    hidden_neurons=None,
    hidden_activation='tanh',
    skip_connection=False,
    n_ensemble=50,
    n_estimators=6,
    max_samples=256,
    contamination=0.1,
    random_state=None,
    device=None
)
```

Ізольований ліс був останнім у пулі основних – базових алгоритмів проектованої розробки у цьому розділі роботи, але на цьому етапі проектування не завершується, оскільки, як видно на загальній схемі узагальненої архітектури після усіх основних – базових алгоритмів у проєкції організовується процес консолідованого рішення на основі ансамблювання по процедурі простого голосування, що і буде описано далі даного розділу проектування.

2.9 Детектування на основі інструкції ансамблювання

Інструкція ансамблювання, загалом, може мати різного виду формати. Зокрема, у цій роботі розглядається ансамблювання на основі простого голосування основних – базових алгоритмів, що були описані вище по тексту роботи бакалавра. Проектування фактично на основі ансамблювання – це фактично створення метаалгоритму для прийняття рішень, які є виділяючими ненормальні дані – аномальні елементи даних. При цьому простому голосуванні типу ансамблювання всі описані основні алгоритми приймають колегіальне рішення – консолідоване рішення зичайною більшістю алгоритмів, що голосують і у проектованій узагальненій архітектурі представляють розгалуження на першому етапі. На цьому закінчимо процес проектування розробки і перейдемо у наступному розділі до етапу практично-прикладного конструювання програмної розробки бакалаврської роботи, що представлена у цій пояснювальній записці..

РОЗДІЛ 3. ПРАКТИЧНО-ПРИКЛАДНЕ КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ НЕ ТИПОВИХ ДАНИХ

У розділі вище було детально зупинено на процесі проектування програмної розробки цієї роботи, а у цьому розділі розглянемо процес практично-прикладного конструювання програмної розробки бакалавра. І першим чином розглянемо, як саме на практиці імплементуються всі необхідні програмні засоби для проектування проекту на основі скриптів технології пайтон, а саме, які інструменти із анаконди дистрибутиву використовуватимуться при реалізації.

3.1 Уведення бібліотек та фреймворків пайтона із дистрибутиву анаконди

Практична реалізація розпочинається із імпортування всіх необхідних програмних інструментів для написання коду у всіх скриптах розроблюваного проекту. Нище наведено практичну реалізацію, яка саме із анаконди дистрибутиву імпортуються бібліотеки і які саме бібліотеки для цього потрібні у скриптах.

```
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from pyod.models.knn import KNN
from pyod.models.lof import LOF
from pyod.models.pca import PCA as pyod_PCA
from pyod.models.ocsvm import OCSVM
from pyod.models.ifoorest import IForest
```

Як видно основними інструментами є програмні класи із бібліотеки пайоді, яка в бекенді використовує скікітлерн. Відносяться такі класи пайоді: головних компонент метод, найближчих сусідів метод, локальний викид метод, ізольований ліс алгоритм, а також для графічної візуалізації матплотліб. Також, вже стандартно використовуються маніпуляційні даними пайндас та нумпай бібліотеки. Перейдемо покроково до практичного конструювання програмної реалізації на практиці рішень. І першим кроком розглянемо зчитування даних із зовні у скрипт пайтона даного проекту.

3.2 Уведення даних нерухомості із зовнішнього середовища

Зчитування даних реалізовується на практиці маніпуляційною бібліотекою пайндас і зокрема методом цсв для читання довільних текстових файлів із довільними розширеннями та роздільниками між числами та цілими частинами чисел. Цей метод має багато гіперпараметрів для свого управління, зокрема, це гіперпараметри для вибору символів на розділення цілої і дробової частин, розділення числових значень між собою у текстовому документі, а також наявний гіперпараметр для вказування індексу рядка у якому знаходяться назви змінних, тобто назви колонок фрейму, як структури даних пандас. Кодове представлення наведено нище і має вид, який наведений далі по тексту.

```
"""Read data"""
df = pd.read_csv("./ASSESSMENT_COST_OF_HOUSING/ASSESSMENT_COST_OF_HOUSING.csv",
                 delimiter=";", decimal=".", header=0)
```

Після практичної реалізації процесу зчитування даних у фреймову пандасівську структуру даних вони візуалізуються за допомогою методу плот, або окремої функції принт. У скрипті використовується стандартна пайтонівська

функція print для виводу містимого різних типів та структур змінних і результат має вигляд, що зображений у табличному виді далі у цьому потоці тексту.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
1	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.9	9.14	21.6
2	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.9	5.33	36.2

Рис.3.1 – Приклад частки множини вхідних даних для прикладно-практичного завдання

Рядки у двовимірному масиві – це об’єкти (квартири, будинки, тощо) нерухомості ринку, а колонки двовимірного масиву – це атрибути об’єктів (квартири, будинки, тощо) нерухомості ринку.

Загалом дані для роботи були вибрані з інтернет платформи Кагл. Ці дані представляють параметри ряду об’єктів нерухомості пошуку локації міста Бостон, що в США. Елемент даних – нерухомість представляється точкою у висьмивірному просторі даних, або що є еквівалентним – вектором у восьмовимірному просторі даних, тобто всі дані нерухомості – це вектори у восьмовимірному просторі у яких початки співпадають, а кінці рознесені у векторному просторі. У висьмивірному векторному просторі – декартовому координатному просторі вектори згущаються в окремих локаціях, тобто утворюють кластерну структуру. Ізольовані вектори, тобто ті вектори у яких кінці є ізольованими точками не можливо для попереднього розгляду представити, оскільки – це багатовимірний простір. Відомо, що найпершим кроком для ідентифікації аномальних даних на прикладі нерухомості необхідно попередньо візуалізувати дані в декартовій системі координат, щоб розглянути чи є аномальні дані. Для візуалізації багатовимірних даних використовують найрізноманітніші підходи, як кластерного аналізу, наприклад,

самолорганізуючі карти Кохонена так і інші алгоритми. Проте у даній роботі для візуалізації восьмивимірного простору в якому перебувають об'єкти нерухомості у роботі використовується метод із матричних факторизацій, а саме, використовується головних компонент метод із гіперпараметром кількість компонент дорівнює дві. Можна загалом використовувати найрізноманітніші алгоритми, але автором, як розробником було прийнято рішення використати головних компонент метод. Розгляд практичного використання на основі нього програмного коду описано у наступному підрозділі.

3.3 Зображення багатовимірних даних на основі PCA

Для початку тут по тексту наводиться кодове представлення фрагменту скрипта загалом проекту для візуалізації топологічної структури восьмивимірного простору даних об'єктів нерухомості при проектуванні їх у двовимірний простір, що можна бачити нижче у практичному кодовому записі нище по даному тексту.

```

"""Apply of PCA-transformation for visualization of data in two-dimensional space"""
def plot(x1, x2, x3):
    #Initialization 'PCA'
    pca = PCA(n_components = 2)
    #Fitting 'PCA'
    pca.fit(x1)
    #Transformation
    df_transformed = pca.transform(x1)
    #Visualization of PCA-transformed data in two-dimensional space
    plt.plot(df_transformed[:,0], df_transformed[:,1], 'o', color=x2)
    plt.xlabel('pc1')
    plt.ylabel('pc2')
    plt.title('Visualization of PCA-transformed ' + x3)
    plt.grid()
    plt.show()

x1, x2, x3 = df, 'blue', 'data'
plot(x1, x2, x3)

```

Як видно з кодового представлення візуалізація організована у вигляді створеної функції. При цьому використовуються дві компоненти PCA із скітлерн бібліотеки – головних компонент метод та методи навчання і метод трансформації даних. Також використовується матплотліб бібліотека із рядом методів для графічної візуалізації вже після трансформації даних до двовимірного декартового простору. По координатним вісям відкладаються головні компоненти, які представляють собою метаінтерпритовані ознаки сукупності даних після методу трансформації головних компонент алгоритму. Для графічних символів, надписів використовуються різні методи матплотліб бібліотеки, які представлені у фрагменті коду вище.

Далі перейдемо до кодового подання по черзі всіх алгоритмів, що були наведені у узагальненій архітектурі при проектуванні програмної реалізації і першим підходом нище бути описано підхід на основі статистичних тестів.

3.4 Практичне конструювання детектування на основі інструкцій статистичних критеріїв

Одним із добре і давно відомих підходів для виявлення аномальних даних, зокрема, об'єктів нерухомості є використання різного виду статистичних тестів. Тут у роботі при практичному конструюванні програмного забезпечення використовується безпосередньо статистичний тест Грабба, який реалізується на основі центрування даних і їх нормування на основі середньоквадратичного відхилення. Вибираючи інтервал довіри, як правило плюс, мінус три сигма з довірчою імовірністю нуль цілих дев'яносто вісім сотих вибирається нульова статистична гіпотеза, що дані – об'єкт нерухомості із фіксованим значенням

характеристики не значимо відрізняється від решта об'єктів нерухомості, тобто топологічно не ізольований, а значить не представляє собою аномальні дані, тобто для нього не використовується гіпотеза аш один. Якщо ж значення перевищує довірчий інтервал, то спрацьовує гіпотеза аш один і тоді об'єкт нерухомості із таким фіксованим значенням характеристики є викидом у статистичному розумінні, а значить він є аномальним і можливо збільшення ризиків при його покупці. Разом з тим, заради справедливості, відзначимо, що недоліком є те, що лише по одній характеристиці можна визначити об'єкт даних, а не інтегрально по всіх разом. Нище наведено кодове представлення статистичного тесту Грабба, що і реалізовано у даному практичному проекті цієї роботи бакалавра.

```
""" #1. Outliers and inliers detection based on statistical test"""
#Inliers
inliers_1 = df[((df-df.mean())/df.std()).abs()<3].dropna()
#Outliers
outliers_1 = df[((df-df.mean())/df.std()).abs()>3].dropna(how='all')

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_1, 'blue', 'inliers_1'
plot(x1, x2, x3)
```

У наступному підрозділі буде наведено практичне конструювання наступного основного алгоритму детектування аномальних об'єктів нерухомості на основі методу найближчих сусідів. Зупинемося на ньому.

3.5 Практичне конструювання детектування на основі інструкцій найближчих сусідів

Практично-прикладне конструювання другого основного алгоритму реалізується на метричному підході, тобто на основі метричних векторних

просторів. Використовується програмна компонента в основі із скітлерну – найближчих сусідів алгоритм. Кодове представлення наведено нище по тексту.

```
""" #2. Outliers and inliers detection based on k-Nearest Neighbors (kNN) """
#Initialization
knn = KNN(n_neighbors=5, metric='minkowski', n_jobs=-1)
#Fit
knn.fit(df)

#Binary labels (0: inliers, 1: outliers)
outliers_inliers_2 = knn.labels_
#Outlier scores. The higher, the more abnormal. Outliers tend to have higher scores
outlier_scores_2 = knn.decision_scores_

#Inliers
inliers_2 = df[outliers_inliers_2==0]
#Outliers
outliers_2 = df[outliers_inliers_2==1]

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_2, 'blue', 'inliers_2'
plot(x1, x2, x3)

#Apply of PCA-transformation for visualization of 'outliers' in two-dimensional space
x1, x2, x3 = outliers_2, 'red', 'outliers_2'
plot(x1, x2, x3)
```

Як видно з кодового напису у цьому алгоритмі обрані наступні значення для гіперпараметрів. Метрика відстані була вибрана Мінковського, оскільки з практично-еврістичного досвіду відомо, що у переважній кількості прикладів вона спрацьовує найкраще. Значення гіперпараметру для кількості сусідів обрано число п'ять, це було встановлено також із практично-еврістичного досвіду застосування даного алгоритму на практиці. Звісно, що з метою оптимізації практичної роботи даного алгоритму на практиці у розумінні мінімізації затрати часу на роботу процесорних ядер виконувалося його розпаралелення по логічним процесорам, тобто потокам. У даній роботі було обрано числове значення мінус одиниця, що тлумачиться, як використання всіх фізичних процесорних ядер комп'ютера на якому реалізовуватиметься конструйований софт.

Далі на практиці застосовується метод фіттинг – підгонка або ще його називають навчання найближчих сусідів алгоритму, а після того за допомогою атрибуту лейбелс повертаються мітки для кожного рядка фрейму пандас. Атрибут лейбелс повертає значення нуля для елементів викидів та значення одиниці для

елементів внутрішніх у топологічному розумінні даних, тобто нормальних даних – нормальних об'єктів нерухомості. Також цей алгоритм повертає значення функції аномальності для всіх даних фрейму, тобто функція аномальності – це аналог вагової функції, тобто наскільки елемент даних – нерухомість є аномальною чи не аномальною. Нище наведено графічно частковий вид результату аномальних даних, які повертає найближчих сусідів алгоритм.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV		0
18	0.80271	0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2	0	19.4656
25	0.84054	0	8.14	0	0.538	5.599	85.7	4.4546	4	307	21	303.42	16.51	13.9	1	24.6228
27	0.95577	0	8.14	0	0.538	6.047	88.8	4.4534	4	307	21	306.38	17.28	14.8	2	22.032
32	1.38799	0	8.14	0	0.538	5.95	82	3.99	4	307	21	232.6	27.71	13.2	3	16.6251
34	1.61282	0	8.14	0	0.538	6.096	96.9	3.7598	4	307	21	248.31	20.34	13.5	4	19.881
48	0.25387	0	6.91	0	0.448	5.399	95.3	5.87	3	233	17.9	396.9	30.81	14.4	5	16.9563
54	0.0136	75	4	0	0.41	5.888	47.6	7.3197	3	469	21.1	396.9	14.8	18.9	6	16.868
74	0.07896	0	12.83	0	0.437	6.273	6	4.2515	5	398	18.7	394.92	6.78	24.1	7	18.6967
102	0.22876	0	8.56	0	0.52	6.405	85.4	2.7147	5	384	20.9	70.8	10.63	18.6	8	20.1464
118	0.13058	0	10.01	0	0.547	5.872	73.1	2.4775	6	432	17.8	338.63	15.37	20.4	9	16.4326

Рис.3.2 – Аномал. дані та їх оцінки аномал. на основі процедури найближ. сусідів

На цьому у роботі завершується практично-прикладне конструювання першого основного алгоритму детектування даних – найближчих сусідів алгоритму. Тому перейдемо до наступного основного алгоритму, що представлений в узагальненій архітектурі програмної реалізації детектування аномальних об'єктів нерухомості.

Така візуалізація даних, що представляють ненормальні об'єкти нерухомості, тобто аномальні є достатньо наочною для працівників ріелторських контор, оскільки як видно методи пандас для візуалізації аномальних нерухомостей представлена градієнтом кольору.

Ріелтори, як працівники агенцій нерухомості легко можуть користуватися градієнтом кольору для виявлення високоризикових об'єктів нерухомості і цим

самим виконувати консалтинг більш якісно для своїх клієнтів інформуючи їх про можливі ризики, що можуть виникнути при покупці. Перейдемо до наступного основного – базового алгоритму програмного забезпечення.

3.6 Практичне конструювання детектування на основі інструкції LOF

Наступним серед основних – базових алгоритмів конструюванню підлягає кодування один елемент зовні. Цей метод також застосовується інтегрально до всіх атрибутів даних – характеристик нерухомості інтегрально і це його перевага над статистичними тестами, як критеріями для детектування нестандартних об'єктів нерухомості – аномальних даних. Також до його переваг слід віднести розпаралелюваність його роботи по фізичним ядрам компютера, що відповідно прискорює його виконання у часі і з вищим коефіцієнтом корисної дії.

```
""" #3. Outliers and inliers detection based on Local Outlier Factor (LOF) """
#Initialization
lof = LOF(n_neighbors=20, metric='minkowski', n_jobs=-1)
#Fit
lof.fit(df)

#Binary labels (0: inliers, 1: outliers)
outliers_inliers_3 = lof.labels_
#Outlier scores. The higher, the more abnormal. Outliers tend to have higher scores
outlier_scores_3 = lof.decision_scores_

#Inliers
inliers_3 = df[outliers_inliers_3==0]
#Outliers
outliers_3 = df[outliers_inliers_3==1]

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_3, 'blue', 'inliers_3'
plot(x1, x2, x3)

#Apply of PCA-transformation for visualization of 'outliers' in two-dimensional space
x1, x2, x3 = outliers_3, 'red', 'outliers_3'
plot(x1, x2, x3)
```

Він також відноситься до класу метричних алгоритмів на кшталт найближчих сусідів, але він на практиці використовує, як видно з коду двадцять найближчих сусідів. Також він має для свого управління гіперпараметр, який задає відстань між даними у метричному векторному просторі. При конструюванні розробки використовувалася відстань Мінковського. Після застосування фітінгового методу використовувався атрибут міток, який повертає двовимірний фрейм із трьох колонок, перша з яких вказує індекси рядків даних, тобто ту нерухомість, які топологічно утворюють кластер, інша аномальні нерухомості, тобто топологічно ізольовані точки. Також є колонка із значеннями функції, яка зображає вагові значення належності даних – нерухомості до аномалій. Таке практичне конструювання дає змогу задавати апріорний поріг значень по якому машина може приймати рішення щодо тієї чи іншої нерухомості стосовно того чи вона аномальна чи не аномальна. Нижче наведено частку практичних результатів даного методу.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV		0
18	0.80271	0	8.14	0	0.538	5.456	36.6	3.7965	4	307	21	288.99	11.69	20.2	0	1.16151
25	0.84054	0	8.14	0	0.538	5.599	85.7	4.4546	4	307	21	303.42	16.51	13.9	1	1.15067
27	0.95577	0	8.14	0	0.538	6.047	88.8	4.4534	4	307	21	306.38	17.28	14.8	2	1.02668
30	1.13081	0	8.14	0	0.538	5.713	94.1	4.233	4	307	21	360.17	22.6	12.7	3	0.971491
32	1.38799	0	8.14	0	0.538	5.95	82	3.99	4	307	21	232.6	27.71	13.2	4	0.977425
33	1.15172	0	8.14	0	0.538	5.701	95	3.7872	4	307	21	358.77	18.35	13.1	5	0.974008
34	1.61282	0	8.14	0	0.538	6.096	96.9	3.7598	4	307	21	248.31	20.34	13.5	6	1.07532
48	0.25387	0	6.91	0	0.448	5.399	95.3	5.87	3	233	17.9	396.9	30.81	14.4	7	1.17113
54	0.0136	75	4	0	0.41	5.888	47.6	7.3197	3	469	21.1	396.9	14.8	18.9	8	1.29767
95	0.12204	0	2.89	0	0.445	6.625	57.8	3.4952	2	276	18	357.98	6.65	28.4	9	1.04937

Рис.3.3 – Аномал. дані та їх оцінки аномал. на основі процедури LOF

З вихідних даних та функції, яка зображає вагові значення належності нерухомості до аномалій чи ні і дає можливість м'якого детектування ненормальних об'єктів нерухомості. У випадку, якщо потрібне чітке детектування нестандартних нерухомостей, то апріорі задається порогове значення вагової функції аномальності і відбувається відсікання елементів по фіксованому апріорному значенню заданого

порогу. Тому дане практична реалізація коду передбачає в спроектованій архітектурі реалізувати на практиці чітку та мяку детекцію аномальних нерухомостей у даній практично-конструйованій розробці у вигляді програмного коду. Перейдемо до наступного основного – базового підходу для ідентифікації ненормальних нерухомостей – виявленню нестандартних – аномальних нерухомостей.

3.7 Практичне конструювання детектування на основі інструкції PCA

З архітетктури видно, що двічі застосовується головних компонент алгоритм. Перший раз він використовувався при практичному конструюванні блоку візуалізації багатовимірних даних – за допомогою нього практично реалізовувалося проектування нерухомостей – даних із восьмивимірному простору у двовимірний після того ці дані графічно зображувалися у двовимірному просторі даних.

```

""" #4. Outliers and inliers detection based on Principal Component Analysis (PCA) """
#Initialization
pc = pyod_PCA(n_components=7)
#Fit
pc.fit(df)

#Binary labels (0: inliers, 1: outliers)
outliers_inliers_4 = pc.labels_
#Outlier scores. The higher, the more abnormal. Outliers tend to have higher scores
outlier_scores_4 = pc.decision_scores_

#Inliers
inliers_4 = df[outliers_inliers_4==0]
#Outliers
outliers_4 = df[outliers_inliers_4==1]

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_4, 'blue', 'inliers_4'
plot(x1, x2, x3)

#Apply of PCA-transformation for visualization of 'outliers' in two-dimensional space
x1, x2, x3 = outliers_4, 'red', 'outliers_4'
plot(x1, x2, x3)

```

У цьому конкретному прикладі цей алгоритм використовується для детектування нерухомостей, що не подібні на інші по своїх фічах. Він так само, як і вище наведені основні алгоритми повертає за допомогою атрибуту класу нормальні об'єкти нерухомості та аномальні об'єкти нерухомості, а також функцію ваги аномальності. Чим більше модульне значення функції ваги аномальності тим більше належить нерухомість до ненормальних даних. Нище наведено частину фіч та даних, які повернув цей основний – базовий алгоритм нерухомості, які вважаються аномальними у цьому датасеті, що приводився раніше у роботі.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV		0
55	0.01311	90	1.22	0	0.403	7.249	21.9	8.6966	5	226	17.9	395.93	4.81	35.4	0	292.521
57	0.01432	100	1.32	0	0.411	6.816	40.5	8.3248	5	256	15.1	392.9	3.95	31.6	1	228.804
142	3.32105	0	19.58	1	0.871	5.403	100	1.3216	5	403	14.7	396.9	26.82	13.4	2	308.39
152	1.12658	0	19.58	1	0.871	5.012	88	1.6102	5	403	14.7	343.28	12.12	15.3	3	342.093
155	3.53501	0	19.58	1	0.871	6.152	82.6	1.7455	5	403	14.7	88.01	15.02	15.6	4	343.512
162	1.83377	0	19.58	1	0.605	7.802	98.2	2.0407	5	403	14.7	389.61	1.92	50	5	292.795
163	1.51902	0	19.58	1	0.605	8.375	93.9	2.162	5	403	14.7	388.45	3.32	50	6	246.573
195	0.01381	80	0.46	0	0.422	7.875	32	5.6484	4	255	14.4	394.23	2.97	50	7	289.203
196	0.04011	80	1.52	0	0.404	7.287	34.1	7.309	2	329	12.6	396.9	4.08	33.3	8	386.846
197	0.04666	80	1.52	0	0.404	7.107	36.6	7.309	2	329	12.6	354.31	8.61	30.3	9	283.068

Рис.3.4 – Аномал. дані та їх оцінки аномал. на основі процедури PCA

На цьому опис даного основного базового алгоритму головних компонент завершимо практичне конструювання і перед тим, як перейти до наступного базового алгоритму для ідентифікації ненормальних об'єктів нерухомості, а саме однокласового алгоритму векторів опорних напишемо ще декілька коментарів щодо керування роботою алгоритму компонент головних. Важливість використання цього алгоритму ще обумовлена й тим фактом, що він дає мета інтерпритацію фіч нових даних у новому координатному просторі. Як видно з останнього графічного зображення дані також мають градієнт кольору, а це означає, що такий дизайн полегшує роботу працівників агенції нерухомості щодо відзначення нестандартних нерухомостей і приділити їм увагу в плані зменшення ризиків, які працівник агенції

нерухомості може сповістити клієнту, який планує її купити і цим самим його застерегти щодо виникнення у нього потенційних ризиків при покупці фіксованої нерухомості, а значить більш об'єктивне рішення прийняти. Перейдемо до наступного основного – базового алгоритму, який наявний у архітектурі.

3.8 Практичне конструювання детектування на основі інструкції OCSVM

Даний алгоритм підпрограма конструюється також на основі використання програмних інструментів бібліотеки пайоді, зокрема використання класу OCSVM.

```
""" #5. Outliers and inliers detection based on One-class SVM"""
#Initialization
svm = OCSVM(kernel='rbf', nu=0.5, gamma='auto')
#Fit
svm.fit(df)

#Binary labels (0: inliers, 1: outliers)
outliers_inliers_5 = svm.labels_
#Outlier scores. The higher, the more abnormal. Outliers tend to have higher scores
outlier_scores_5 = svm.decision_scores_

#Inliers
inliers_5 = df[outliers_inliers_5==0]
#Outliers
outliers_5 = df[outliers_inliers_5==1]

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_5, 'blue', 'inliers_5'
plot(x1, x2, x3)

#Apply of PCA-transformation for visualization of 'outliers' in two-dimensional space
x1, x2, x3 = outliers_5, 'red', 'outliers_5'
plot(x1, x2, x3)
```

Загалом практична кодова реалізація при конструюванні принципово не відрізняється від попередніх кодових реалізацій основних алгоритмів архітектури. Він також використовує метод фіттинг і після того на основі атрибуту лебл повертає фрейм із трома колонками. Перша колонка – це індекси рядків для нормальних

нерухомостей, тобто неізолюваних точок протору об'єктів нерухомості, друга колонка – це індекси рядків ненормальних нерухомостей, тобто ізолюваних точок, які представляють собою аномальні нерухомості набору даних. Також з нище наведеного рисунку видно градієнт кольору, який полегшує працівникам ріелторської фірми виявляти підозрілі нерухомості і відзначати клієнтам-покупцям щодо виникнення потенційних ризиків купівлі нерухомості.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV	
2	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	0 -6.88877e...
6	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311	15.2	395.6	12.43	22.9	1 -0.000106...
7	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311	15.2	396.9	19.15	27.1	2 0.000393538
10	0.22489	12.5	7.87	0	0.524	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15	3 0.000150099
17	0.7842	0	8.14	0	0.538	5.99	81.7	4.2579	4	307	21	386.75	14.67	17.5	4 -9.93217e...
21	0.85204	0	8.14	0	0.538	5.965	89.2	4.0123	4	307	21	392.53	13.83	19.6	5 -6.35043e...
29	1.00245	0	8.14	0	0.538	6.674	87.3	4.239	4	307	21	380.23	11.98	21	6 0.000378195
37	0.08014	0	5.96	0	0.499	5.85	41.5	3.9342	5	279	19.2	396.9	8.77	21	7 0.000461045
48	0.25387	0	6.91	0	0.448	5.399	95.3	5.87	3	233	17.9	396.9	30.81	14.4	8 -0.000119...
58	0.15445	25	5.13	0	0.453	6.145	29.2	7.8148	8	284	19.7	390.68	6.86	23.3	9 -9.68218e...

Рис.3.5 – Аномал. дані та їх оцінки аномал. на основі процедури OCSVM

Перейдемо до практичного конструювання наступного алгоритму архітектури для виявлення аномальних нерухомостей у датасеті нерухомостей.

3.9 Практичне конструювання детектування на основі інструкції IForest

Цей програмний алгоритм ізолюваного лісу практично конструюється на основі елементів – класу пайоді бібліотеки і підтримує у бекенді функції бібліотеки скітлерн, оскільки вона використовується разом із пайоді через API і застосовуючи

метод фітінгу, а потім атрибуту лебл повертає аналогічно зазначеному внутрішні, зовнішні точки у векторному просторі даних – об'єктів нерухомості, а також вагову функцію аномальності для кожного рядка даних датасету. Кодове представлення таке, як представлено далі із коментарями до кожного блоку скрипта пайтона.

```
""" #6. Outliers and inliers detection based on IsolationForest"""
#Initialization
forest = IForest(n_estimators=100, max_samples='auto', contamination=0.1,
                 max_features=1.0, bootstrap=True, n_jobs=-1)

#Fit
forest.fit(df)

#Binary labels (0: inliers, 1: outliers)
outliers_inliers_6 = forest.labels_
#Outlier scores. The higher, the more abnormal. Outliers tend to have higher scores
outlier_scores_6 = forest.decision_scores_

#Inliers
inliers_6 = df[outliers_inliers_6==0]
#Outliers
outliers_6 = df[outliers_inliers_6==1]

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_6, 'blue', 'inliers_6'
plot(x1, x2, x3)

#Apply of PCA-transformation for visualization of 'outliers' in two-dimensional space
x1, x2, x3 = outliers_6, 'red', 'outliers_6'
plot(x1, x2, x3)
```

Нище зображено частину фіч та рядків аномальних нерухомостей, які повернув ізольований ліс конструйованого програмного забезпечення.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV		0
54	0.0136	75	4	0	0.41	5.888	47.6	7.3197	3	469	21.1	396.9	14.8	18.9	0	-0.0717258
142	3.32105	0	19.58	1	0.871	5.403	100	1.3216	5	403	14.7	396.9	26.82	13.4	1	-0.116882
143	4.0974	0	19.58	0	0.871	5.468	100	1.4118	5	403	14.7	396.9	26.42	15.6	2	-0.0926154
144	2.77974	0	19.58	0	0.871	4.903	97.8	1.3459	5	403	14.7	396.9	29.29	11.8	3	-0.0886747
145	2.37934	0	19.58	0	0.871	6.13	100	1.4191	5	403	14.7	172.91	27.8	13.8	4	-0.0923755
146	2.15505	0	19.58	0	0.871	5.628	100	1.5166	5	403	14.7	169.27	16.65	15.6	5	-0.10248
147	2.36862	0	19.58	0	0.871	4.926	95.7	1.4608	5	403	14.7	391.71	29.53	14.6	6	-0.0992685
152	1.12658	0	19.58	1	0.871	5.012	88	1.6102	5	403	14.7	343.28	12.12	15.3	7	-0.058948
154	1.41385	0	19.58	1	0.871	6.129	96	1.7494	5	403	14.7	321.02	15.12	17	8	-0.0155179
155	3.53501	0	19.58	1	0.871	6.152	82.6	1.7455	5	403	14.7	88.01	15.02	15.6	9	-0.0744504

Рис.3.6 - Аномал. дані та їх оцінки аномал. на основі процедури ізольований ліс

3.10 Практичне конструювання детектування на основі інструкції ансамблювання

У цьому підрозділі практично конструється кодове рішення консолідованого рішення усіх разом взятих базових алгоритмів архітектури, що була спроєтована.

```
""" #7. Outliers and inliers detection based on Ensemble of Outliers Detectors
(simple voting of Ensemble of Outliers Detectors) """
#Create list of binary arrays
out_int = [outliers_inliers_2, outliers_inliers_3, outliers_inliers_4,
           outliers_inliers_5, outliers_inliers_6]

#Sum of binary arrays
outliers_inliers_7 = np.zeros(shape=(len(df),))
for i in out_int:
    outliers_inliers_7 += i

#Simple voting of Ensemble of Outliers Detectors
for i, j in enumerate(outliers_inliers_7):
    if j >= 3:
        outliers_inliers_7[i] = 1
    else:
        outliers_inliers_7[i] = 0

#Inliers
inliers_7 = df[outliers_inliers_7==0]
#Outliers
outliers_7 = df[outliers_inliers_7==1]

#Apply of PCA-transformation for visualization of 'inliers' in two-dimensional space
x1, x2, x3 = inliers_7, 'blue', 'inliers_7'
plot(x1, x2, x3)

#Apply of PCA-transformation for visualization of 'outliers' in two-dimensional space
x1, x2, x3 = outliers_7, 'red', 'outliers_7'
plot(x1, x2, x3)
```

Цей процес консолідованого рішення усіх базових алгоритмів базується на основі їх загального голосування по процедурі простого голосування. У цьому випадку всі базові алгоритми приймають рішення з однаковою вагою і консолідоване, загальне рішення приймається більшістю голосів, тобто більшістю голосів базових алгоритмів. Кожен із алгоритмів має свій апріорі заданий поріг значення аномальності нерухомості. Консолідоване рішення базових алгоритмів зменшує ризики пропущення підозрілих об'єктів нерухомості, що у свою чергу

зменшує потенційні ризики для клієнтів, які планують купувати нерухомість. Консолідованість дає змогу на рівні різних характеристик нерухомості та метарівні характеристик нерухомості ідентифікувати ненормальні нерухомості. Загальне рішення ансамблювання алгоритмів зображено далі.

Index	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
48	0.25387	0	6.91	0	0.448	5.399	95.3	5.87	3	233	17.9	396.9	30.81	14.4
54	0.0136	75	4	0	0.41	5.888	47.6	7.3197	3	469	21.1	396.9	14.8	18.9
145	2.37934	0	19.58	0	0.871	6.13	100	1.4191	5	403	14.7	172.91	27.8	13.8
146	2.15505	0	19.58	0	0.871	5.628	100	1.5166	5	403	14.7	169.27	16.65	15.6
155	3.53501	0	19.58	1	0.871	6.152	82.6	1.7455	5	403	14.7	88.01	15.02	15.6
156	2.44668	0	19.58	0	0.871	5.272	94	1.7364	5	403	14.7	88.63	16.14	13.1
165	2.924	0	19.58	0	0.605	6.101	93	2.2834	5	403	14.7	240.16	9.81	25
196	0.04011	80	1.52	0	0.404	7.287	34.1	7.309	2	329	12.6	396.9	4.08	33.3
197	0.04666	80	1.52	0	0.404	7.107	36.6	7.309	2	329	12.6	354.31	8.61	30.3
199	0.0315	95	1.47	0	0.403	6.975	15.3	7.6534	3	402	17	396.9	4.56	34.9

Рис.3.7 – Аномал. дані та їх оцінки аномал. на основі ансамблювання

Як видно результуючі дані також мають кольоровий градієнт і цим самим полегшують візуально бачити працівнику ріелторської фірми підозрілі нерухомості, тобто, аномальні дані.

На цьому процес практичного конструювання створеної архітектури програмного рішення завешено і перейдемо до наступного викладу матеріалу цієї кваліфікаційної бакалаврської праці.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Аварії з викидом радіоактивних речовин

Аварія з викидом (розливом) радіоактивних речовин – аварія на радіаційно небезпечному об'єкті, яка спричинила викид (розлив) радіоактивних речовин за межі встановлених захисних бар'єрів, і (чи) потужність дози іонізуючого випромінювання перевищує встановлені норми і загрожує довкіллю [10, 11].

Радіаційне забруднення – забруднення поверхні землі, атмосфери, води, продуктів харчування, харчової сировини, кормів та різних предметів радіоактивними речовинами у кількості, яка перевищує рівень, встановлений стандартами, нормами і правилами радіаційної безпеки [10, 11].

Виникнення аварій (катастроф) на транспорті, підприємствах та енергетичних установках з викидом радіоактивних речовин становить серйозну небезпеку.

Фактори небезпеки викиду (розливу) радіоактивних речовин – забруднення навколишнього середовища, небезпека для всього живого, що опинилося на забрудненій місцевості (загибель людей, тварин, знищення посівів та ін.), крім того, внаслідок можливого атомного вибуху виникнення сильних руйнувань на значній території.

Пропонуємо заходи, необхідні кожній людині, яка може опинитися у районі можливої аварії з викидом (розливом) радіоактивних речовин. Досвід ліквідації аварій на АЕС показав високу ефективність цих заходів.

Дії у випадку загрози виникнення аварії з викидом (розливом) радіоактивних речовин [10, 11]:

Сирени і переривчасті гудки підприємств та транспортних засобів - це сигнал "Увага всім". Негайно ввімкніть приймачі радіотрансляційної мережі, радіоприймач

або телевізор. Уважно слухайте інформацію про надзвичайну ситуацію та інструкції про порядок дій. Не користуйтеся без потреби телефоном, щоб він був вільним для зв'язку з вами. При оголошенні небезпечного стану не поспішайте! Зберігайте спокій. Часу для ретельного виконання заходів захисту у вас достатньо. Попередьте сусідів, надайте допомогу інвалідам, дітям та людям похилого віку. Проведіть заходи щодо зменшення проникнення радіаційних речовин у квартиру (будинки): щільно закрийте вікна та двері, щілини заклейте. Підготуйте запас питної води: наберіть воду у ємності, що закриваються, підготуйте найпростіші засоби санітарної обробки (наприклад, мильний розчин для обробки рук), перекрийте крани. Підготуйтеся до можливої евакуації: підготуйте документи, цінності та гроші, предмети першої необхідності, ліки, мінімум білизни та одягу, запас консервованих продуктів на 2-3 доби. Речі упакуйте та укладіть у найбільш захищеному від проникнення зовнішнього забруднення приміщенні. Дізнайтеся в органах державної влади та місцевого самоврядування місце збору мешканців для евакуації. Від'єднайте всі споживачі електричного струму від електромережі, вимкніть газ.

Дії під час раптової аварії з викидом (розливом) радіоактивних речовин [10]:

Зберігайте спокій, уникайте паніки. З одержанням повідомлення (по радіо або іншому каналу оповіщення) про радіаційну небезпеку, виконайте вищезазначені заходи. Негайно укрийтеся в житлових будинках. Стіни дерев'яного будинку послаблюють іонізуюче випромінювання в 2 рази, цегляного — у 10 разів, заглиблені укриття (підвали) з покриттям із дерева — у 7 разів, з покриттям із цегли або бетону у 40 - 100 разів. Слухайте повідомлення органів цивільної оборони. Уточніть час початку евакуації. Попередьте сусідів про початок евакуації. Проведіть заходи щодо зменшення проникнення радіаційних речовин в квартиру (будинки). Підготуйте запас питної води, найпростіші засоби санітарної обробки. Проведіть йодну профілактику. Йодистий калій вживати після їжі разом з чаєм, соком або водою 1 раз на день протягом 7 діб; дітям до двох років — по 0,040 г на один прийом; дітям від двох років та дорослим — по 0,125 г на один прийом. Водно-спиртовий розчин йоду приймати після їжі 3 рази на день протягом 7 діб; дітям до двох років — по 1-2 краплі

5% настоянки на 100 мл молока (консервованого) або году-вальної суміші; дітям від двох років та дорослим — по 3-5 крапель на склянку молока або води. Наносити на поверхню кінцівок рук настоянку йоду у вигляді сітки 1 раз на день протягом 7 діб. Швидко зберіть потрібні документи, цінності, ліки, продукти та інші необхідні речі.

Підготуйтеся до евакуації [10, 11]:

Зберіть документи, цінності та гроші, предмети першої необхідності, ліки, білизну, одяг, запас консервованих продуктів на 2-3 доби. Речі упакуйте та зберігайте у найбільш захищеному від проникнення зовнішнього забруднення приміщенні. Надайте допомогу дітям, інвалідам та людям похилого віку. Вони підлягають евакуації в першу чергу. По можливості негайно залишіть зону радіоактивного забруднення. Перед виходом з будинку вимкніть джерела електро-, водо- і газопостачання, візьміть підготовлені речі, надіньте протигаз (респіратор, ватно-марлеву пов'язку), верхній одяг (плащ, пальто, накидку), гумові чоботи. З прибуттям на нове місце, проведіть дезактивацію засобів захисту, одягу, взуття та санітарну обробку шкіри на спеціально обладнаному санітарному пункті або ж самостійно (зняти верхній одяг і, ставши спиною проти вітру, витрясти його); повісити одяг на перекладину чи мотузку, віником або щіткою змести з нього радіоактивний пил та вимити водою; обробити відкриті ділянки шкіри водою або розчином із індивідуального протихімічного пакета (ПП-8), який буде виданий кожному. Для обробки шкіри можна використовувати вату, марлю чи просто рушники. Дізнайтеся у місцевих органів державної влади та місцевого самоврядування адреси організацій, що відповідають за надання допомоги потерпілому населенню.

Необхідні для виконання правила [10, 11]:

Використовуйте для харчування лише консервоване молоко та продукти, що зберігалися у зачинених приміщеннях і не зазнали радіоактивного забруднення. Не пийте молоко від корів, які пасуться на забруднених пасовищах. Не вживайте овочів, які росли на забрудненому радіоактивними речовинами ґрунті. Не пийте воду з відкритих джерел та із мереж водопостачання після офіційного оголошення

радіаційної небезпеки, колодязі накрийте плівкою або кришками. Уникайте тривалого перебування на забрудненій території, особливо на ґрунтових дорогах та на траві, не ходіть до лісу, не збирайте у лісі ягід, грибів та квітів, не купайтесь у водоймах. У приміщеннях, призначених для перебування людей, щодня робіть вологе прибирання, бажано з використанням миючих засобів. Змініть взуття перед входом у приміщення, вимийте його водою або витріть вологою ганчіркою, верхній одяг витрусіть та почистіть вологою щіткою. У разі перебування на відкритій, забрудненій радіоактивними речовинами місцевості, обов'язково використовуйте засоби захисту (особливо під час вітру). Захистіть органи дихання, шкіру та волосся. Засоби індивідуального захисту можна не використовувати в приміщеннях, в тиху погоду без вітру та після дощу. По можливості користуйтеся індивідуальними засобами захисту органів дихання і шкіри промислового виробництва.

4.2 Техніка безпеки при проведенні електричних вимірювань та випробувань установки

1. Загальні вимоги охорони праці.

1.1. Під об'єктом випробувань (вимірювань) слід розуміти один або кілька однотипних об'єктів, які долають (вимірюваних) одночасно одним і тим же засобом випробувань (вимірювань).

1.2. Під випробуваннями (вимірами) обладнання слід розуміти випробування (вимірювання) діючих електроустановок, що знаходяться в експлуатації, а також випробування (вимірювання), що здійснюються при монтажі або ремонті обладнання.

1.3. До проведення вимірювань та випробувань електрообладнання допускається персонал, який пройшов спеціальну підготовку і перевірку знань Правил охорони праці (правил безпеки) при експлуатації електроустановок (далі - Правил) комісією, до складу якої включаються фахівці з випробувань обладнання, мають V групу - в електроустановках напругою понад 1000 В і IV групу - в електроустановках напругою до 1000 В.

1.4. До проведення вимірювань та випробувань електрообладнання допускаються працівники не молодше 18 років, які пройшли попередній медичний огляд і не мають протипоказань до виконання зазначеної роботи.

1.5. Працівник при прийомі на роботу проходить вступний інструктаж. Перед допуском до самостійної роботи працівник повинен пройти:

- Навчання за програмами підготовки за професією;
- Первинний інструктаж на робочому місці;
- Перевірку знань інструкцій: з охорони праці, з надання першої допомоги потерпілим від нещасних випадків на виробництві, з пожежної безпеки, щодо застосування засобів захисту, необхідних для безпечного виконання робіт.

1.6. Для виробничого навчання працівнику має бути надана термін, достатній для ознайомлення з обладнанням, апаратурою, оперативними схемами і одночасного вивчення необхідної для даної посади нормативної і технічної літератури.

1.7. До роботи з електровимірювальними приладами повинні допускатися працівники, які пройшли інструктаж і навчання безпечним методам праці, перевірку знань правил та інструкцій відповідно до займаної посади стосовно до виконуваної роботи з присвоєнням відповідної групи з електробезпеки і не мають медичних протипоказань.

1.8. Працівників, які суміщають професії, навчають і інструктують за правилами безпеки праці в повному обсязі з їх основної та сумісництвом професій (посад).

1.9. Допуск до самостійної роботи оформляється відповідним розпорядженням по структурному підрозділу підприємства.

1.10. Новоприйнятому працівникові видається кваліфікаційне посвідчення, в якому має бути зроблений відповідний запис про перевірку знань інструкцій і правил та про право на виконання спеціальних робіт.

1.11. Кваліфікаційне посвідчення для чергового персоналу під час виконання службових обов'язків може зберігатися у начальника зміни цеху або при собі відповідно до місцевих умов.

1.12. Працівники, які не пройшли перевірку знань у встановлені терміни, до самостійної роботи не допускаються.

1.13. Працівник в процесі роботи зобов'язаний проходити:

- Повторні інструктажі;
- Перевірку знань інструкції з охорони праці та діючої інструкції по наданню першої допомоги потерпілим при нещасних випадках на виробництві один раз на рік;
- Медичний огляд один раз на рік;
- Перевірку знань Правил для працівників, що мають право підготовки робочого місця, допуску, право бути виконавцем робіт, наглядачами або членом бригади, один раз на рік.

1.14. Працівники, які отримали незадовільну оцінку при кваліфікаційній перевірці, до самостійної роботи не допускаються і не пізніше одного місяця повинні пройти повторну перевірку.

1.15. Право на проведення вимірювань та випробувань підтверджується записом в рядку «Свідectво на право проведення спеціальних робіт» посвідчення про перевірку знань норм і правил роботи в електроустановках.

1.16. Виконавець робіт, зайнятий випробуваннями електрообладнання, а також працівники, які проводять випробування одноосібно з використанням стаціонарних випробувальних установок, повинні пройти місячне стажування під контролем досвідченого працівника.

2. Вимоги до організації та підготовці випробувань і вимірювань.

2.1. Вимірювання та випробування слід проводити за програмами і методиками, технічними умовами організацій-виробників або стандартам на продукцію. Вимірювання та випробування електрообладнання або електроустановок, нововведених в експлуатацію, проводяться відповідно до норм, передбачених діючими Правилами улаштування електроустановок (ПУЕ), вимогами організацій-виробників, вимог стандартів, а вимірювання і випробування діючих електроустановок та електрообладнання - в обсязі вимог норм і правил їх експлуатації. Випробувальні установки (електролабораторії) повинні бути зареєстровані в органах Держенергонагляду.

2.2. Дозвіл на проведення вимірювань та випробувань діючих електроустановок повинно бути оформлено у відповідності з діючими Правилами охорони праці. Вимірювання та випробування, що проводяться на ІС за програмами і методиками, проводять без оформлення будь-якого розпорядження, поза ними - за розпорядженням керівника вимірювань або випробувань. Розпорядження заноситься в спеціальний журнал.

2.3. Випробування і вимірювання в діючих електроустановках напругою понад 1000 В провадиться за нарядом.

Випробування і вимірювання електричних двигунів і різного окремо стоїть одиничного обладнання напругою вище 1000 В, від яких від'єднані струмопровідні частини (кабелі живлення, шини) і заземлені, можуть виконуватись за розпорядженням.

2.4. Допуск за нарядами та розпорядженнями на проведення вимірювань та випробувань проводиться тільки після виводу з робочих місць інших бригад, які працюють на підлягає випробуванню або вимірюванню обладнанні, і здачі ними нарядів або повідомлення про закінчення робіт за розпорядженням.

2.5. До складу бригад, які проводять випробування або вимірювання, можуть бути включені працівники з числа ремонтного персоналу з групою з електробезпеки не нижче II для виконання підготовчих робіт, охорони випробовується, а також для роз'єднання і з'єднання шин, жил кабелю, проводів. Ремонтний персонал, включений до складу бригади, до початку випробувань або вимірювань повинен бути проінструктований виконавцем робіт про заходи безпеки при випробуваннях або вимірах. До складу бригади, що здійснює монтаж або ремонт обладнання для проведення випробувань і (або) вимірювань, можуть бути включені працівники з числа персоналу налагоджувальних організацій або електролабораторій. В цьому випадку випробуваннями і (або) вимірами керує виконавець робіт або за його вказівкою старший працівник з групою з електробезпеки не нижче IV з числа персоналу електролабораторії або налагоджувальної організації.

2.6. Підготовку об'єкта і засобів вимірювання до випробувань або вимірювань слід проводити за відсутності на них напруги і залишкового заряду. Робоча напруга і залишковий заряд повинні бути також зняті з інших об'єктів (інших частин об'єктів випробувань і вимірювань), якщо не виключено дотик або наближення до них, або ці об'єкти повинні бути на час підготовки та проведення випробувань огорожені.

2.7. Складання та розбирання випробувальних і (або) вимірювальних ланцюгів слід виконувати при відсутності на об'єкті випробування і (або) вимірювання або його частини і на засобах вимірювання та (або) випробування напруги і залишкового заряду.

2.8. Збірку кола випробування (вимірювання) обладнання виробляє персонал бригади, яка проводить випробування (вимірювання). При цьому слід виконати захисне і робоче заземлення випробувальної або вимірювальної установки і при

необхідності - захисне заземлення корпусу обладнання. При приєднанні випробувальної або вимірювальної установки до мережі напругою 380/220 В на виводі високої напруги установки слід встановити заземлення. Перетин мідного заземлюючого провідника повинен бути не менше 4 кв. мм. Корпус пересувної випробувальної установки має бути заземлений окремим заземлювальним провідником з гнучкого мідного проводу перерізом не менше 10 кв. мм. Перед випробуванням необхідно перевірити надійність заземлення корпусу.

2.9. Знімати заземлення, накладені в електроустановки і перешкоджають проведенню випробувань або вимірювань, і накладати їх знову слід тільки за вказівкою керівника випробувань або вимірювань.

2.10. Провід між випробовуваним обладнанням і випробувальною установкою спочатку повинен бути приєднаний до її заземленого висновку високої напруги. Цей провід слід закріплювати так, щоб уникнути наближення (підсікання) до знаходяться під напругою струмоведучих частин.

ВИСНОВКИ

1. Зпроектовано програмне забезпечення для детектування аномальних об'єктів нерухомості на основі зпроектованого метаалгоритму детекції, що дало змогу зменшити ризики клієнтів щодо шахрайських об'єктів нерухомості.

2. Практично зконструйовано програмне забезпечення для детектування аномальних об'єктів нерухомості на основі зпроектованого метаалгоритму детекції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pastukh O., Stefanyshyn V., Baran I., Yakymenko I., Vasylyuk V. Mathematics and software for controlling mobile software devices based on brain activity signals // The International Workshop on Information Technologies: Theoretical and Applied Problems
2. Pastukh O., Petryk M., Bachynskiy M., Mudryk I., Stefanyshyn V. Processing of Cerebral Cortex Neurosignals from EEG Sensors and Recognizing Specific Types of Mechanical Movements Elements of Patient Limbs under the Cognitive Feedback Influences // International Workshop on Computer Information Technologies in Industry 4.0. 2023. №3468. С. 61–70.
3. Yasniy O., Pastukh O., Didych I., Yatsyshyn V., Chykhira I. Application of machine learning for modeling of 6061-T651 aluminum alloy stress– strain diagram // Procedia Structural Integrity. 2023. №48. С. 183-189.
4. Pandas documentation. URL: <https://pandas.pydata.org/docs/> (дата звернення: 12.04.2024)
5. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design // Scientific Journal of TNTU. 2023. №1. С. 54–65.
6. Pastukh O., Petryk M., Bachynskiy M., Mudryk I., Stefanyshyn V. Processing of Cerebral Cortex Neurosignals from EEG Sensors and Recognizing Specific Types of Mechanical Movements Elements of Patient Limbs under the Cognitive Feedback Influences // International Workshop on Computer Information Technologies in Industry 4.0. 2023. №3468. С. 61–70.
7. Pastukh O., Petryk M., Bachynskiy M., Mudryk I., Stefanyshyn V. Processing of Cerebral Cortex Neurosignals from EEG Sensors and Recognizing Specific Types of Mechanical Movements Elements of Patient Limbs under the Cognitive Feedback Influences // International Workshop on Computer Information Technologies in Industry 4.0. 2023. №3468. С. 61–70.

8. Matplotlib 3.8.3 documentation. URL: <https://matplotlib.org/stable/index.html>
(дата звернення: 12.04.2024)
9. Пастух О.А., Яцишин В.В., Жаровський Р.О., Шаблій Н.Р. Software tool for productivity metrics measure of relational database management system // Математичне моделювання. 2023. №1 (48). С.7-17.
10. Мелех Л.В. Безпека життєдіяльності та охорона праці : Навчальний посібник. Львів: Львівський державний університет внутрішніх справ. 2022. 219 с.
11. Сокурєнко В.В. Безпека життєдіяльності та охорона праці : Підручник. Харків: Харків. нац. ун-т внутр. справ. 2021. 308 с.

ДОДАТКИ

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра програмної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПІ
_____ Петрик М.Р.
"___" _____ 2024 р

РОЗРОБКА СИСТЕМИ ДЛЯ ІДЕНТИФІКАЦІЇ АНОМАЛЬНИХ ДАНИХ

ТЕХНІЧНЕ ЗАВДАННЯ

Вид робіт:

Кваліфікаційна робота

на здобуття освітнього ступеня "Бакалавр"
спеціальність 121 "Інженерія програмного забезпечення"

"УЗГОДЖЕНО"

керівник кваліфікаційної роботи

_____ д.т.н., проф. Пастух О.А.

"___" _____ 2024 р.

"ВИКОНАВЕЦЬ"

студент групи СП-41

_____ Антонюк Д.Т.

"___" _____ 2024 р.

Тернопіль 2024

ЗМІСТ

1. Підстави до розробки
2. Призначення до розробки
3. Вимоги до програмної системи
 - 3.1. Функціональні характеристики
 - 3.2. Склад та параметри технічних засобів
 - 3.3. Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

1. ПІДСТАВИ ДО РОЗРОБКИ

Розробка програмної системи виконується у відповідності до графіку навчального плану на 2023 – 2024 н.р. та згідно наказу ректора Тернопільського національного технічного університету імені Івана Пулюя на виконання кваліфікаційної роботи на здобуття освітнього ступеня бакалавр.

Тема кваліфікаційної роботи на здобуття освітнього ступеня бакалавр: ”Розробка системи для ідентифікації аномальних даних” призначена для створення програмного інструменту у сфері виявлення аномальних даних.

2. ПРИЗНАЧЕННЯ РОЗРОБКИ

Програмна система – ”Розробка системи для ідентифікації аномальних даних” призначена для агенцій нерухомості. Дана ситема може виявляти аномальні дані у довільних наборах даних.

В основу програмного забезпечення покладено алгоритми машинного навчання для виявлення аномальних даних у наборах даних. Наприклад, виявлення об'єктів нерухомості, що представляють собою заманухи на ринку нерухомості довільної локації.

3. ВИМОГИ ДО ПРОГРАМНОЇ СИСТЕМИ

3.1. Функціональні характеристики

- формувати та візуалізувати дані аномальних об'єктів нерухомості на ринку нерухомості
- формувати та візуалізувати величину довірчої імовірності щодо аномальності об'єкту даних
- оцінювати точність прийнятих рішень

3.2. Склад та параметри технічних засобів

1) персональний комп'ютер з 16 Гб оперативної пам'яті, встановленою системою Windows 8, 8.1, 10. Не менше 200 Мб вільного місця на жорсткому диску. Двоядерний процесор із двома потоками та тактовою частотою від 1.6 GHz і більше.

2) наявність встановленого дистрибутиву Анаконди та ядра мови програмування Python із бібліотеками Pandas, Scikit-learn, Numpy.

3.3. Інформаційна та програмна сполучність

Програмна система повинна коректно функціонувати в операційній системі Windows різних поколінь. Програмні модулі програмної системи повинні працювати з високою степінню надійності та точності. Розробку виконувати з використанням бібліотек та технологій мови Python в середовищі програмування Spyder.

4. СТАДІЇ РОЗРОБКИ

В ході реалізації роботи програмна система повинна пройти крізь стадії життєвого циклу розробки програмного забезпечення:

- аналіз предметної області
- проектування архітектури програмної системи
- практичне конструювання програмної системи
- реалізація певних елементів візуалізації даних
- тестування якості роботи окремих програмних модулів програмної системи
- оформлення супровідної документації
- здача роботи

5. ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для розробленої програмної системи для детектування аномальних даних на ринку об'єктів нерухомості повинні бути розроблені наступні документи:

- пояснювальна записка
- технічне завдання
- презентаційний матеріал
- додатки

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблена програмна система має виконувати всі вимоги, що складаються з перерахованих у даному технічному завданні.

Приймання проводиться спеціально створеною екзаменаційною комісією у відведені терміни, що вказані у календарному плані завдання кваліфікаційної роботи на здобуття освітнього ступеня бакалавр.

Тези доповіді на конференції

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



Студентське наукове
товариство



VII МІЖНАРОДНА
 студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
НАУКИ.

АКТУАЛЬНІ ПИТАННЯ"

25-26 квітня 2024 р.

(збірник тез конференції)

Тернопіль 2024

Марків К. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КРУЇЗНОЇ КОМПАНІЇ З ВИКОРИСТАННЯМ SPRING FRAMEWORK	331
Музика В. АСИНХРОННІСТЬ ПРОГРАМУВАННЯ В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	332
Петришин Я., Марцинюк Я. РОЛЬ SI/CD У ПІДВИЩЕННІ ЕФЕКТИВНОСТІ ТА НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	333
Бурса В., Мудрик І. АВТОМАТИЗОВАНА СИСТЕМА ОЦІНКИ РІВНЯ ЗНАННЯ ІНОЗЕМНОЇ МОВИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ	335
Антонюк Д., Пастух О. АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНСАМБЛЮВАННЯ АЛГОРИТМІВ	336
Чорна Х., Пастух О. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУ ПАСАЖИРОПЕРЕВЕЗЕНЬ	337
Бабинець К., Пастух О. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТОВОЇ ІНФОРМАЦІЇ	338
Бойко Д., Пастух О. ПРОГРАМА ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО КРЕДИТУВАННЯ	339
Яковів Б., Пастух О. ПРОГРАМНА СИСТЕМА ДЛЯ НЕВРОЛОГІЧНОЇ ДІАГНОСТИКИ	340
Гашинський Р., Мельник Н. АВТОМАТИЗОВАНИЙ АНАЛІЗ СХЕМИ ДОКУМЕНТІВ З ЧАСТКОВО-СТРУКТУРОВАНИМИ ДАНИМИ	341

УДК 629.735.072.8.08:004(043.2)

Антонюк Д. – ст. гр. СП-41, Пастух О.

Тернопільський національний технічний університет імені Івана Пулюя

АРХИТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНСАМБЛЮВАННЯ АЛГОРИТМІВ

Науковий керівник: д.т.н., професор Пастух О.А.

Antonyuk D., Pastukh O.

Ternopil Ivan Puluj National Technical University

SOFTWARE ARCHITECTURE BASED ON ASSEMBLING ALGORITHMS

Supervisor: Dr.Sc., Professor Pastukh O.A.

Доволі часто при вирішенні складних завдань практики не вдається досягнути високих метрологічних характеристик у роботі програмного забезпечення. Така ситуація має місце при детектуванні аномальних даних у наборі даних. У такому разі використовують ансамблювання алгоритмів, оскільки це потенційно дає змогу покращити метрологічні характеристики програмного забезпечення. У роботі пропонується архітектура програмного забезпечення на основі ансамблювання базових алгоритмів з метою покращення покращення метрологічних характеристик для вирішення завдання детектування аномальних даних. Архітектура наведена на рис.1.

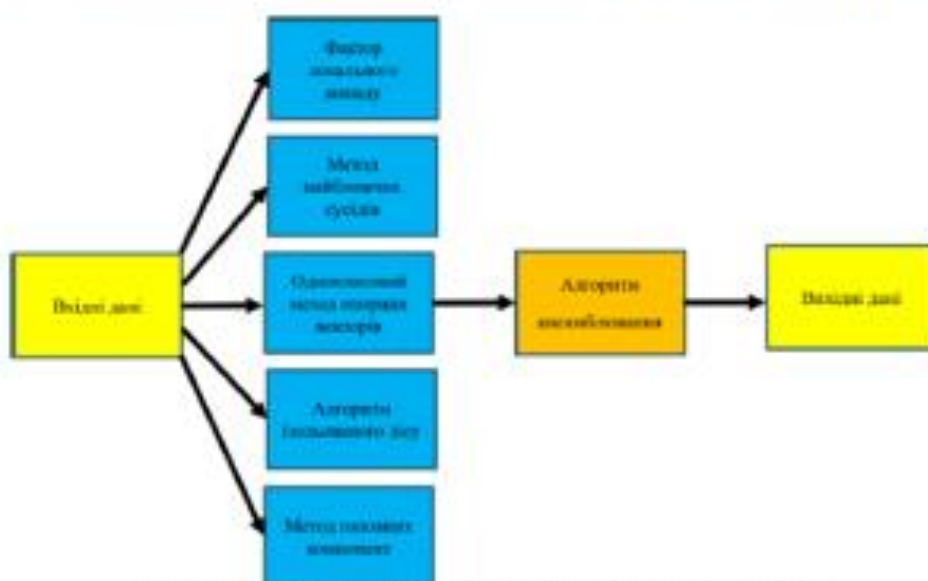


Рис.1 – Ансамблева архітектура програмного забезпечення

На першому ярусі архітектури програмного забезпечення знаходяться базові алгоритми, а на другому ярусі знаходиться мета-алгоритм їх ансамблювання на основі простого голосування базових алгоритмів.